APL/JHU
SR 75-3A
JUNE 1975

See 1473

DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY

APPENDIX A, FINDINGS AND RECOMMENDATIONS OF PREVIOUS STUDIES

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

Approved for public release; distribution is unlimited

APL/JHU
SR 75-3A
JUNE 1975

DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY

APPENDIX A, FINDINGS AND RECOMMENDATIONS OF PREVIOUS STUDIES

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20810
Operating under Contract N00017 72 C 4401 with the Department of the Navy

19. Key Words (cont'd)

Computer Software (cont'd)
  Software Visibility in Weapon System Acquisition
  System Engineering of Computer Software

Computer Systems
  Analysis and Validation of System Requirements
  Computer System Resource Development Plan
  Milestoned Development Plan
  Provisions for Growth in System Requirements
  Standard Criteria for Weapon System Computer Resources
    Acquisition Management
  System Engineering of Computer Systems
  System Integration and Test Capability
  Systems Engineering Agent
  Technical Staffing of Program Manager Organization

Conference Reports, computer
  Air Force Logistics Command Operation Flight Support
  ASAP Ad Hoc Committee for Army Tactical Data System
    Software Development
  Automatic Data Processing in DoD
  CCIP-85
  Electronics-X Study
  Government/Industry Software Sizing and Costing Workshop
  Monterey Symposium on the High Cost of Software
  Proceedings of the Aeronautical Systems Software Workshop
  Project Pacer Flash
  Tactical Computer Software Acquisition and Maintenance
    Staff Study

ABSTRACT

This appendix to the DoD Weapon Systems Software Management Study conducted by APL contains background material extracted and/or summarized from 10 previous DoD-sponsored studies. The studies were designated Baseline Documents by the Department of Defense Software Management Steering Committee and are particularly relevant to the subject of Weapon Systems software. A brief introduction specifying the purpose of each study and a summary of its findings and/or conclusions are included. Recommendations are summarized for each study that provided them. Whenever such study recommendations are available, abbreviated versions of the APL recommendations (from the main report) that correlate most closely are included for reference.

ACKNOWLEDGMENT

This Appendix on the Findings and Recommendations of Previous Studies was prepared by J. M. Park and edited by H. M. Stainer.

TABLE OF CONTENTS

CONTENTS

INTRODUCTION

1. INTRODUCTION

This appendix to the DoD Weapon Systems Software Management Study conducted by APL contains material that has been extracted and/or summarized from 10 previous studies. These studies were selected and designated by the Department of Defense Software Management Steering Committee as Baseline Documents; they are listed in Table 1-1. A more detailed listing is given in Table 1-2 for the reader's convenience.

Baseline Documents are, by definition, studies that are particularly relevant to the subject of Weapon Systems software. Generally, a common theme points to the need to manage Weapon Systems software in a manner that will reduce costs and provide greater visibility throughout the total acquisition cycle.

A brief introduction specifying the purpose of each study and a summary of the findings/conclusions have been included. A summary of recommendations has been included for those studies that provided them. Whenever such study recommendations are available, abbreviated versions of the major APL recommendations that correlate most closely are included for reference.

TABLE 1-1

BASELINE DOCUMENTS -- STUDIES AND WORKSHOPS

Title	Sponsor	Date
Electronics-X: A Study of Military Electronics with Particular Reference to Cost and Reliability	ARPA,DDR&E	Jan 74
Tactical Computer Software Acquisition and Maintenance Staff Study	OSD(I&L)	Oct 73
Army Scientific Advisory Panel Ad Hoc Committee for Army Tactical Data System Software Development	Army	Oct 74
Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85)	AFSC	Apr 72
Project Pacer Flash	AFLC	Sep 73
Automatic Data Processing Costs in the Defense Department	DDR&E	Oct 74
Air Force Logistics Command Operation Flight Program Support	AFLC	Dec 74
Proceedings of the Aeronautical Systems Software Workshop	AFSC	Apr 74
Proceedings of a Symposium on the High Cost of Software Held at the Naval Postgraduate School	AFSC,ARO,ONR	Sep 73
Government/Industry Software Sizing and Costing Workshop	ESD	Oct 74

TABLE 1-2

CITATION LIST OF PREVIOUS STUDIES

1. H. P. Gates, Jr., B. S. Gourary, and S. J. Deitchman, Electronics-X: A Study of Military Electronics with Particular Reference to Cost and Reliability. Volume 2: Complete Report (Unclassified), Institute for Defense Analyses, Arlington, Va., R-195, January 1974.

2. E. T. Reich, Tactical Computer Software Acquisition and Maintenance Staff Study (Unclassified), Office of the Assistant Secretary of Defense, Washington, D.C., October 1973.

3. Report of the Army Scientific Advisory Panel Ad Hoc Committee for Army Tactical Data System Software Development (Unclassified), Army Scientific Advisory Panel, Washington, D.C., October 1974.

4. B. W. Boehm and A. C. Haile, Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85). Executive Summary (Revised Edition) (Unclassified), USAF Space and Missile Systems Organization, Los Angeles, Cal., SAMSO TR 72-122, February 1972.

   Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85). Volume I. Highlights (Unclassified), USAF Space and Missile Systems Organization, Los Angeles, Cal., SAMSO TR 72-141, April 1972.

   The CCIP-85 study also produced ten supporting volumes:

Volume II	Command and Control Requirements: Overview Annex A: Strategic Requirements Annex B: Air Defense Requirements Annex C: Tactical Requirements
Volume III	Command and Control Requirements: Intelligence
Volume IV	Technology Trends: Software
Volume V	Technology Trends: Hardware
Volume VI	Technology Trends: Sensors
Volume VII	Technology Trends: Integrated Design
Volume VIII	Interservice Coordination Trends
Volume IX	Analysis
Volume X	Current Research and Development
Volume XI	Roadmaps

TABLE 1-2 (cont'd)

5. Project Pacer Flash — Volume I. Executive Study and Final Report (Unclassified), Air Force Logistics Command, Wright-Patterson AFB, Ohio, September 1973.

6. D. A. Fisher, Jr., Automatic Data Processing Costs in the Defense Department (Unclassified), Institute for Defense Analyses, Arlington, Va., IDA-P-1046, October 1974.

7. A Report on Air Force Logistics Command Operation Flight Program Support (Unclassified), System Development Corp., Santa Monica, Cal., TM-5439/000/00 and TM-5439/001/00, December 1974.

8. Proceedings of the Aeronautical Systems Software Workshop (Draft) (Unclassified), Air Force Systems Command, Washington, D.C., April 1974.

9. Proceedings of a Symposium on the High Cost of Software Held at the Naval Postgraduate School, Monterey, California, on September 17-19, 1973 (The High Cost of Software) (Unclassified), Stanford Research Institute, Menlo Park, Cal., AD-777-121, SRI Project 3272, September 1973.

10. Summary Notes of a Government/Industry Sizing and Costing Workshop (Unclassified), Electronic Systems Division, Hanscomb AFB, Mass., February 1975.

ELECTRONICS-X

2. ELECTRONICS-X: A STUDY OF MILITARY ELECTRONICS WITH PARTICULAR REFERENCE TO COST AND RELIABILITY

2.1 INTRODUCTION

The Electronics-X study program was conducted by the Institute for Defense Analyses (IDA) with the assistance of representatives of industry, private research organizations, and Government, as well as private consultants. The charter for the study "called for recommendations that could readily be translated into implementable policies, procedures, and practices. The study took into account broad principles recommended by earlier investigations and sought specific data leading to suggested approaches to a reduction of the costs of electronics acquisition and support that would be consistent with the role of military electronics: enhancing the combat capability and crisis readiness of military forces."

Electronics-X "identified problem areas, assessed the magnitude of the problems, attempted to determine their principal causes, and then formulated recommendations for eliminating, as far as possible, those causes. The recommended courses of action are not unique solutions of the problems but rather represent the consensus of best judgments of the Study Group."

"This report is concerned with three kinds of costs, development, production, and support. Empirical evidence suggests that, statistically, production and support costs are positively correlated; but that development effort can be applied to reduce either one or the other or the sum of the two. Because support costs occur in future years and are neither accounted for by the project manager nor paid for out of current funds, the present management emphasis is on holding down just the total of development and production costs, even though lifetime support costs may dominate. Methods to internalize the sum of unit acquisition and support costs to a single responsible party are needed if that sum is to be reduced."

2.2 FINDINGS/CONCLUSIONS

Electronics-X concentrated on five major high-impact areas of the military electronics acquisition process: (a) data collection and feedback, (b) requirements, (c) competition and management options, (d) reliability enhancement, and (e) maintenance training. Software was also considered, as were many other areas; detailed recommendations were made for each area.

This section (2.2) summarizes selected major findings.

2.2.1 Cost Data Collection and Reporting Systems

A profound lack of valid cost data and overwhelming inadequacies in pertinent reporting systems were encountered throughout the Electronics-X study. Specifically, "DoD appears to have no cost accounting system capable of providing data on the full life-cycle costs of any electronic subsystem."

2.2.2 Requirements and Acquisition Decisions

"A requirement for a system or subsystem may be defined as including performance, physical characteristics, cost, quantity, and schedule — all in conformity with a statement of threat or need. While the overall requirements and acquisition decision process includes attention to all these components, the current approach to establishing a requirement tends to start with desired performance and characteristics. Cost, quantity, and schedule are modifiers, added later. Thus, requirements tend to be performance-driven, with inadequate early consideration of pragmatic essentials."

2.2.3 Design to Facilitate Competition

In about two-thirds of military prime contract awards, competition is a missing ingredient. "Even when a program does admit development competition, there is a strong tendency for the Government to become locked into a single supplier in subsequent production. The loss of Government freedom of action permits suppliers to force price up by various devices."

2.2.4 Design for Improved Reliability

"The essence of reliability is simplicity. Empirical evidence indicates clearly that most equipments of high unit production cost or high complexity have lower mean time between failures (MTBF) than equipments of lower unit production cost or lower complexity."

2.2.5 Maintenance Training

Three major factors, high turnover, long training period, and low median level of experience (less than 3 years), result in an expensive and unproductive maintenance force, high training cost, and high turnover of maintenance personnel.

2.2.6 Software

"Software costs have exceeded hardware costs by large factors in some military systems using general-purpose computers."

"Software developments are frequently behind schedule, causing other costs to spiral."

"Software 'unreliability' is a euphemism for software errors."

"The complexity and extent of the software may well be a measure of the mismatch between the hardware and the problem."

"Major sources of excessive software costs in conventional systems employing central uniprocessors are the following:

1. "Selecting hardware and starting programming before the system is designed in detail.
2. "Overburdening the central processor with tasks that can be accomplished by specialized peripherals."
3. Selecting a central processor that is too small with consequent overutilization of the computer, and resorting to poor programming practices.
4. "Program overintegration.
5. "Lack of adequate discipline in software development.
6. "Developing a new high-level programming language for every job.
7. "Starting programming before the computer design is complete."

2.3 RECOMMENDATIONS OF ELECTRONICS-X

This section (2.3) summarizes selected major recommendations from the Electronics-X study.

2.3.1. Cost Data Collection and Reporting Systems

"A systematic effort should be undertaken to develop a stepwise implementation of a complete and uniform cost accounting system throughout DoD, with emphasis on valid input data." The system could be limited in scope initially but must later evolve to cover full costs of both acquisition and support.

2.3.2 Requirements and Acquisition Decisions

"In exploring and establishing a system requirement, give performance, physical characteristics, cost, quantity, and schedule equal status from the beginning and perform tradeoffs among these early" in the process.

2.3.3 Design to Facilitate Competition

The interchangeability of similar equipments intended for similar applications can be accomplished by including (or by requiring prime contractors to include) mechanical, electrical, and environmental interface standards for each unit as a part of military electronic equipment specifications. This process will lay the groundwork for future design and price competition through production and for ready replacement of old designs by new-generation equipment.

2.3.4 Design for Improved Reliability

"Limit the complexity of new subsystem or equipment designs (as measured by criteria such as unit production cost or parts count) to a level consistent with the reliability required by a mission analysis. Require evidence of compliance as a preliminary to DSARC review for electronic subsystems of major systems, and as preliminary to sub-DSARC review for independently developed electronic subsystems."

2.3.5 Maintenance Training

"Develop fully proceduralized job performance aids for use in routine maintenance of new weapons systems and for selected tasks in high-maintenance portions of existing systems."

2.3.6 Software

"Complete the design of the system and the basic program structure in substantial detail before making major commitments to hardware or coding."

"Limit the aggregation of problems to be solved on a central machine; as an alternative, decentralize processing by providing peripheral special-purpose devices (either analog or digital) or separate peripheral general-purpose machines to perform specific separable functions."

"Select a processor of adequate size to permit underutilizing the computer; write highly modular programs; emphasize structure and overall efficiency rather than hardware efficiency alone."

"Use a standard well-established programming language with which programmers are thoroughly familiar. Use the highest level language appropriate to the task at hand, but avoid the unnecessary development of a unique language."

"Defer coding until the computer design is substantially complete and firm, except for that necessary to verify hardware-software design compatibility."

"Use vigorous discipline in software development, such as the top-down Structured-Programming approach."

2.4 CORRELATION WITH APL RECOMMENDATIONS

The Electronics-X recommendations correlate most closely with the following four APL recommendations: MP1, SE1, SE2, and IP2.

1. Analysis and Validation* of System Requirements MP1

   Direct that a comprehensive analysis and definition program be carried out on software (as well as hardware) elements of each new major Weapon System during the Program Validation Phase, prior to approval of Full Scale Development. The software definition should be carried down to the level of subprograms performing major functions.

   Cost estimates for the development and integration of each subprogram should be based on analysis, simulation, modeling, or construction of its principal parts, as called for by its respective newness or criticality.

2. System Engineering of Computer Systems SE1

   For systems involving several distinct functions, require that the system be divided into functional segments in accordance with the operational requirements. Require during the Program Validation Phase that tradeoff analyses be performed for hardware versus software (i.e., hardwired versus programmable functions) and for different computer system architectures.

*"Validation" is used in the context of Program Validation Phase, as opposed to validation/verification of coded computer programs.

3. Provisions for Growth in System Requirements SE2

Provide for growth and change in requirements on Weapon System computer software by identifying parameters that are uncertain or are likely to change in the future and, where possible, specify the probable limits on such changes. Also identify novel environments and uses of new techniques. Require that computer systems be sized to provide for uncertainties and requirement growth.

4. Disciplined Programming IP2

Require that the computer program development contractor apply a highly disciplined set of engineering practices to the detailed design and programming phases of development. This must involve a clear and disciplined set of standards covering program structure, size, control, interface, formal conventions on data base management, and the demonstration that the standards are enforced in parctice.

TACTICAL COMPUTER SOFTWARE ACQUISITION AND MAINTENANCE STAFF STUDY

3. TACTICAL COMPUTER SOFTWARE ACQUISITION AND MAINTENANCE STAFF STUDY

3.1 INTRODUCTION

In 1973, the Office of the Assistant Secretary of Defense for Installation and Logistics (OSD(I&L)) sponsored and conducted a staff study of selected tactical systems.

In its preface to the report, the study team indicated that, in an effort to make weapons more effective in the modern combatant environment, U.S. military forces are increasingly using and becoming committed to digital computers and their associated software. Because software as such is generally not discussed during major system Defense Systems Acquisition Review Council (DSARC) meetings, the study team decided to survey a selected set of Weapon Systems acquisition programs/projects to identify significant problems related to software acquisition and maintenance. The Weapon System set included the DD-963, LHA, PF, F-14, SAM-D, TACFIRE, AEGIS, E-2C, S-3A, F-15, and VAST.

The study effort was based on an exploratory survey. The study team reviewed policies and held discussions concerning the subject with many individuals associated with the programs or projects. The material was then collected and organized for the study report.

In the Findings and Recommendations Section of the Tactical Computer Software Acquisition and Maintenance Staff Study, it was concluded that "there is a marked absence of DoD management policy guidance regarding the use of digital computers and software in vital automated tactical systems" even though tactical software embodies military doctrinal procedures for accomplishing combat functions. Although "the Congress demonstrates a continued interest in the efficient management of all automatic data processing (ADP) resources, the Office of the Military Budget (OMB) and the Office of the Secretary of Defense (OSD) management policy directives do not include those which are integral to the weapons systems." This occurs because senior DoD personnel have not realized the impact of computers and software developments on tactical systems and the costs involved. This lack of awareness is caused by a rapid revolution in electronics and computer technology.

3.2 FINDINGS/CONCLUSIONS

The study team found that in the absence of high level (DoD) management policy in the Weapon Systems software area, several conditions existed or actions took place.

1. During early Southeast Asia operations, certain tactical systems performing the same functions (air control and defense) for the milit ry services were "unable to automatically exchange information because they were developed independently of each other by contractors under separate service programs or projects." As a result of this experience the OSD and Joint Chiefs of Staff (JCS) addressed the lack of adequate digital data interface standards among certain automated military tactical command systems and initiated immediate corrective projects.

2. "In general, system program/project offices became oriented to acquiring electronics hardware and gave little attention to the process of developing the software. This resulted in a lack of separate progress and cost visibility of the software development process, and of software-oriented standards necessary for effective and efficient development efforts -- configuration management, quality assurance, and cost reporting. Some system programs/projects acquired computer programs from contractors when functionally similar programs and support software existed in government software libraries. This resulted from failure to separate certain software from hardware acquisition."

3. "Military services procured a wide variety of tactical computer hardware and languages, which caused the acquisition of a variety of support software (compilers, assemblers) and executive programs. Not only was this software expensive (millions of dollars were involved), but concurrent acquisition impacted adversely on several system programs'/projects' schedules and costs. Further, tactical system engineers became dependent upon computer programmers because of language orientations."

4. "The vital function of tactical software management (maintenance) during the life of a tactical system - incorporating new and revised tactical doctrine procedures into operating forces — was not adequately recognized and tactical software management activities were victims of: late and inadequate program documentation and supporting material; ineffective contractor configuration management; a variety of languages, executive programs, and compilers; multiple million dollar integration facilities and equipment; and involvement in multi-year development programs without adequate government participation."

Limited cost information indicates that tactical software costs for a major tactical system fall in the range from $25 million to over $50 million. It immediately becomes obvious that the total DoD investment in software for all the many major tactical systems is hundreds of millions of dollars.

"Within the DoD, the Comptroller is responsible for the management of automatic data processing resources. However, the DDR&E, ASD(I&L), and ASD(T) are concerned with the acquisition, use, and maintenance of tactical digital computers and software."

3.3 RECOMMENDATIONS OF THE TACTICAL COMPUTER SOFTWARE ACQUISITION AND MAINTENANCE STAFF STUDY

The Secretary of Defense should:

1. "Initiate efforts to make DoD top level management more knowledgeable of the impact of digital computers and software on tactical systems acquisitions and life cycle support, and the costs involved. Along this line, there is a need for clarification and common perspective of the numerous DoD systems — tactical systems, Command and Control Systems, Weapons Systems, and weapons control systems."

2. "Review DoD organizational responsibilities regarding policies and surveillance of tactical digital computer software acquisition, use and maintenance."

3. "Issue policies which will promote more effective and efficient tactical software acquisition, use and maintenance. These should address requirements for standard computers, languages and dialects, executive programs, and support software; greater use of existing computer libraries; separation of software from hardware acquisition when appropriate; requirements for operational user control of computer program development continuously from inception to delivery; and other matters."

"The DoD Materiel Specifications and Standards Board should establish a software panel to provide more adequate guidance for military standards and specifications to support software development and life cycle management. The panel should address documentation, configuration management, quality assurance, cost reporting, and other matters."

3.4 CORRELATION WITH APL RECOMMENDATIONS

The recommendations of the Tactical Computer Software Acquisition and Maintenance Staff Study correlate most closely with the following three APL recommendations: MP2, IP1, and MS3.

1. Software Visibility in Weapon System Acquisition MP2

   Increase the visibility and understanding of major software components of Weapon Systems by putting them on a par with hardware components. This is to be done in terms of configuration control items, DSARC reviews, design reviews, and other aspects of acquisition management.

2. Software Development Support Tools and Facilities IP1

   Ensure that the Full Scale Development program includes provision for adequate modern support tools and facilities, including such items as assemblers, compilers, editors, debugging aids, data base and library management systems, and associated operating systems. Require maximum use of existing proven tools and facilities. Provide that any tools and facilities that will be required by the Operational Support (Maintenance) Agent for system maintenance be delivered in transferable form and also be capable of application to future Weapon System programs.

3. Software Operational Support Agent MS3

   Require that the Software Operational Support (Maintenance) Agent be identified and consulted during the Program Validation Phase to support the Program Manager in providing for maintenance support requirements. Require that the agent be included at the beginning of and throughout Full Scale Development to plan for system integration, testing, and transfer from developmental to operational status.

ASAP REPORT ON ARMY TACTICAL DATA SYSTEM SOFTWARE DEVELOPMENT

4. REPORT OF THE ARMY SCIENTIFIC ADVISORY PANEL AD HOC COMMITTEE FOR ARMY TACTICAL DATA SYSTEM SOFTWARE DEVELOPMENT

4.1 INTRODUCTION

"An ad hoc group of the Army Scientific Advisory Panel (A.S.A.P.) was organized to study Army Tactical Data System software developments and to recommend on actions which will provide improved software at lower cost and in shorter time in future tactical systems."

"The group was first convened on 13 June 1973 and in that and subsequent meetings received information from Army agencies, from other government activities, and from commercial organizations on the history of prior developments, on requirements for the future, and on new hardware and software technologies."

The study group defined the scope of their study as being to determine "the factors that lead to extensive and complex software and to problems in developing software for tactical data systems, and to recommend practices and useful exploratory efforts to mitigate those difficulties." This definition by the study group was somewhat different from the original statement of work in that the statement called for the determination of exploratory development efforts.

4.2 FINDINGS/CONCLUSIONS

As the study began and discussions were started, it became clear "that the solution of tactical system software problems should not be approached solely through software or programming research and development, although certain efforts in that area will prove fruitful." The software problem often originates from a lack of initial system engineering in the overall Tactical Data System that includes such problems as selection of hardware before the problem is clearly defined, provision of a system structure that does not satisfy the problem requirements, and "instinctive and arbitrary choice of conventional central uniprocessors when multiprocessors. a federated system, associative processors, or special purpose processors might be a superior choice." If the systems engineering job is poor, the result is usually a waste of extensive and unwarranted software. "Often development of the software itself is inadequately managed," the kind of discipline normally exercised in hardware development is not present in software development; the problem is not well defined before the programming is started; the tasks are improperly designated, assigned, and monitored; interfaces between program segments are not formally established; and implementation is accomplished in a haphazard manner, rather than being structured.

"The ad hoc study group's investigations led to findings and recommendations in four major areas:

1. System Design and System Hardware

2. Software Design and Development

3. R&D Related to Software

4. Army Management of Software Development."

4.3 RECOMMENDATIONS OF THE ARMY SCIENTIFIC ADVISORY PANEL AD HOC COMMITTEE FOR ARMY TACTICAL DATA SYSTEM SOFTWARE DEVELOPMENT

4.3.1 System Design and System Hardware

The study group recommends an orderly system design, "considering all reasonable alternative system architectures before a development is initiated. . . . Specifically, it is recommended that:

1. Alternative system architectures to that based on a large general purpose computer be evaluated in detail.

2. The system design emphasize the processes to be performed in the tactical environment before defining the processors to be employed.

3. Existing computer systems with standard system software be evaluated before considering the development of new hardware which will require new system software.

4. Hardware and software be defined and developed interactively starting with a definition of the language to be used and the operating system parameters required.

5. Hardware capacity be specified with adequate allowance for a safety factor to reduce the difficulties of programming."

4.3.2 Software Design and Development

"Problems in software design usually result from attempts to obtain very efficient programs to be run on minimum size hardware. Often the program writing must be accomplished without adequate system software tools." The study group "recommends:

1. Early design (or selection) of system software programs and software testing tools.

2. Standardization, specification, documentation and use of a higher-level programming language for Army Tactical Data Systems.

3. Selection and documentation of program libraries of standard tactical operating systems, and of operational tactical program segments with proper consideration of applicable commercial software.

4. Application of the principles of structured programming in tactical data system design.

5. Use of an outside system advisor to assist in program development."

4.3.3 R&D Related to Software

"A number of subjects recommended for research and exploratory development have been identified. The research and exploratory development for Army Tactical Data Systems must be conducted and coordinated in a manner to offer maximum responsiveness to PM ARTADS. Specific studies recommended are:

1. Continued development and evaluation of the standard programming language for Army Tactical Data Systems taking advantage of commercial developments.

2. Development of standard operating systems for tactical applications.

3. Development of computer architectures optimized to the tactical problem, to the standard language, and to means for optimum program development.

4. Improved methods for specifying, selecting, developing, testing and evaluating tactical hardware and software."

4.3.4 Army Management of Software Development

"The study group supports the efforts of PM ARTADS to develop and apply improved management techniques in tactical software development. The following recommendations on management of software development are discussed:

1. Meaningful and realistic tools for management and documentation of software developments.

2. Early agreement on programming language and operating system requirements.

3. Agreement by all parties on software tasks and software specifications.

4. Evaluation of software development progress related to satisfying operational requirements.

5. Specifications and tools for software testing.

6. The requirement for accepting evolutionary development of software in tactical systems."

4.4 CORRELATION WITH APL RECOMMENDATIONS

The recommendations from the Army Scientific Advisory Panel report correlate most closely with the following eight APL recommendations: MP1, MP2, AP1, SE1, SE2, IP2, MS2, and TT1.

1. Analysis and Validation* of System Requirements MP1

   Direct that a comprehensive analysis and definition program be carried out on software (as well as hardware) elements of each new major Weapon System during the Program Validation Phase, prior to approval of Full Scale Development. The software definition should be carried down to the level of subprograms performing major functions.

   Cost estimates for the development and integration of each subprogram should be based on analysis, simulation, modeling, or construction of its principal parts, as called for by its respective newness or criticality.

2. Software Visibility in Weapon System Acquisition MP2

   Increase the visibility and understanding of major software components of Weapon Systems by putting them on a par with hardware components. This is to be done in terms of configuration control items, DSARC reviews, design reviews, and other aspects of acquisition management.

*"Validation" is used in the context of Program Validation Phase, as opposed to validation/verification of coded computer programs.

3. Milestoned Development Plan AP1

Define the requirements for milestones in the Full Scale Development phase to ensure the proper sequence of analysis, design, implementation, integration, test, and review processes. Also, define criteria that will be used to demonstrate that each milestone has been achieved.

4. System Engineering of Computer Systems SE1

For systems involving several distinct functions, require that the system be divided into functional segments in accordance with the operational requirements. Require during the Program Validation Phase that tradeoff analyses be performed for hardware versus software (i.e., hardwired versus programmable functions) and for different computer system architectures.

5. Provisions for Growth in System Requirements SE2

Provide for growth and change in requirements on Weapon System computer software by identifying parameters that are uncertain or are likely to change in the future and, where possible, specify the probable limits on such changes. Also identify novel environments and uses of new techniques. Require that computer systems be sized to provide for uncertainties and requirement growth.

6. Disciplined Programming IP2

Require that the computer program development contractor apply a highly disciplined set of engineering practices to the detailed design and programming phases of development. This must involve a clear and disciplined set of standards covering program structure, size, control, interface, formal conventions on data base management, and the demonstration that the standards are enforced in practice.

7. Systems Engineering Agent MS2

Establish a policy that, for major new Weapon System programs, the Program Manager engage a Systems Engineering Agent to assist in problems arising in the translation of system requirements into detailed hardware and computer system design requirements. The agent, whether Government or contractor, should be highly experienced in system operational requirements, special purpose system hardware, and computer system software and hardware.

8. Software Test Tools TT1

Support development of improved software test and validation tools to reduce the cost and time involved in software verification. These should include automated tools to identify and exercise all branches, to detect and isolate design faults, and to categorize error sources.

CCIP-85

5. INFORMATION PROCESSING/DATA AUTOMATION IMPLICATIONS OF AIR FORCE COMMAND AND CONTROL REQUIREMENTS IN THE 1980s (CCIP-85)

5.1 INTRODUCTION

Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980s (CCIP-85), an Air Force Systems Command (AFSC) Development Planning Study, was completed in early 1972. The study's "purpose was to construct an integrated Air Force R&D program for the 1970's which will develop the information processing technology needed to meet the likely Air Force command and control (C&C) information processing requirements of the 1980's." The primary interest of the study was concerned with "the command and control for the Air Force combatant units."

In order to establish a point of reference in time, it is apparent that the most logical step would be to observe and analyze significant trends in the development of information processing technology. It was stated that information processing is barely adequate to support Air Force C&C functions today. Further, the problems exist in the software area rather than in computer hardware technology; i.e., "the technology of transforming broad functional C&C requirements into specific, detailed and unexceptional sequences of commands for the computer hardware to execute." The study revealed a number of trends that, by the 1980s, will (a) "Make C&C considerably more important to Air Force roles and operations," (b) "Make C&C much more dependent on information processing technology," and (c) "Sharply increase the strains on software technology imposed by C&C requirements." It is apparent that some of the above trends have a high degree of visibility but others are obscure. "But, together, they are gathering momentum from domestic and international pressures and from mutual reinforcement."

Of the three major trends cited above, the third, "strains on software technology," has more relevance to the DoD Software Study because its primary concern is computer software and its ability to meet the Air Force C&C requirements. "Except for airborne functions, the Air Force does not need now, and will not need in the 1980's, the largest, fastest computer hardware available to support C&C operations. . . . However, both now and even more in the 1980's, Air Force C&C will place greater demands on software technology than will other applications. In addition to the growing demands caused by increasingly large data bases, sensor input volume, and user traffic, and the added range and sophistication of C&C decision aids, three unique factors of C&C software will continue to stand out: (1) it must operate in a highly changeable and unpredictable environment, (2) it must operate in a hostile environment, and (3) critical outages or mistakes would affect national survival rather than just costs or the safety of individuals."

5.2 FINDINGS/CONCLUSIONS

The study group produced a list of problems arranged generally in priority sequence as being critical, significant, or appreciable. In the following list from the "Highlights" volume, the first five problems were classified as critical, the next five were significant, and the remainder were appreciable.

"Further, major requirements/R&D mismatches exist within the information processing field. Of the five most critical problem areas identified . . ., four primarily involve software technology needs. Seven of the 15 top problem areas primarily involve software (four primarily hardware, four about equal parts of each). Yet, only about 30 percent of the Air Force information-processing R&D budget is devoted to software technology."

5.2.1 System Design/Exercise Technology

'The nation's survival and prestige rest continuously on the assumption that incidents similar to the Pueblo and Liberty incidents would not occur during grave strategic confrontations. Information processing techniques could and should be doing more in the areas of C&C system requirements analysis, system design, and system exercising to assure that this will not happen."

5.2.2 Software/System Certification

"The Air Force implicitly provides a guarantee to the nation that there are no errors in its command and control software that might escalate a crisis situation or seriously degrade performance during a crisis. Current software technology does not provide the highest possible confidence to back up that guarantee."

5.2.3 Software Timeliness and Flexibility

"Software development is on the critical path in the development of overall Air Force command and control systems. Resulting slippages of six to 12 months in system delivery are typical; often, serious compromises in software flexibility are made to prevent further slippages."

5.2.4 Computer Hardware Survivability

"While the hardware technology forecast reveals no serious mismatch in hardware speed and capacity, there is a serious problem with nuclear hardness. The most serious symptom of this problem is the threat to strategic missiles, which could create an unfavorable strategic asymmetry."

5.2.5 Data Security

"Plans for future Air Force command and control systems assume that this problem will be solved. Current technology provides no assurance that it will."

5.2.6 High-Capacity Airborne Computers

"A comparison of the CCIP-85 analysis of strategic C&C information-processing requirements with the CCIP-85 hardware technology forecast indicates that airborne computers of sufficient speed, size, and hardness for a 1985-era airborne command post will not exist without a dedicated R&D effort."

5.2.7 Multisource Data Fusion

"The TIPI (Tactical Intelligence Processing and Interpretation) system currently under development will provide an initial step toward a capability for fusion of data from many sources into useful information. To exploit this framework properly in the long run, more fundamental studies are necessary to develop and evaluate advanced automated aids to the fusion process."

5.2.8 Communications Processing

"This study supports and reinforces the findings of the MCT Mission Analysis that command and control operational requirements will be significantly degraded if research and development for communications processing are not increased."

5.2.9 Source Data Automation

"Dynamic force management is as susceptible to the 'garbage in, garbage out' phenomenon as any other information processing activity. Advanced computer technology has the potential for providing considerably improved source data reliability and accuracy, as well as improved data acquisition speed, cost, weight, and volume factors."

5.2.10 Image Processing

"Research and development on mission-oriented image-processing functions such as change detection, outline recognition, and semiautomated aids to photointerpreters can yield near-term incremental improvements in C&C capabilities, as well as a base of data and insights for more fundamental future studies."

5.2.11 Computer System Performance Analysis

"Additional R&D efforts would not only pay for themselves in savings; they would also provide significant contributions to software certification and data security assurance."

5.2.12 Associative/Parallel Processor Exploitation

"Particularly for sensor data processing, parallel computer architectures give indications of major potential performance benefits."

5.2.13 Software Transferability

"Two inevitable trends are the continuing increase in C&C software inventory and the eventual upgrading of C&C computer hardware, with its attendant conversion problems. For current perspective, it will take 200 programmers working for three years (or 600 man-years) to convert SAC's software to the new WWMCCS machine."

5.2.14 Computer-Aided Instruction in Computing

"Many problems can be alleviated by increasing the awareness among Air Force personnel of the capabilities and limitations of information processing technology. A modest program in this area could achieve appreciable benefits."

5.2.15 Hardware Destructibility

"Increasingly, aircraft carrying computer hardware containing highly sensitive data will be used in operations outside the United States. Hardware destructibility capabilities must be developed to assure that such data do not fall into enemy hands."

5.3 RECOMMENDATIONS OF CCIP-85

The study group provided some recommendations that are included in the following list. It is apparent that the general trend for recommended future improvements should be directed to the use of new and improved programming techniques, better management of simulation, certification, and production facilities, and to set goals to improve the various factors that affect software and its use.

5.3.1 Structured Programming

1. "Examine, through a series of comprehensive experiments, the use of structured programming by a variety of skill levels in a variety of applications. These efforts must be carefully controlled and fully instrumented, in order to collect data necessary for drawing confident conclusions."

2. "Investigate other methods of bringing structure to the programming process, ranging from establishment of extensive program quality standards to more sophisticated techniques of 'software engineering.'"

3. "Develop career paths and associated training programs and retention incentives for both commissioned and civilian personnel, allowing career advancement in technical disciplines associated with information processing."

5.3.2 Operational Simulation

"Capabilities for operational simulation should be expanded, initially on an experimental basis, for exercising and testing command and control systems. Particularly, semiautomated aids should be developed for scenario generation, script generation, monitoring, and analysis of operational simulations. It is further recommended that research be undertaken toward live, operational simulation of future systems, primarily as a tool for requirements analysis and comparative study. Such a capability can be established by altering and coordinating existing hardware, software, and personnel into a representation of future systems. As with the operational simulation capability itself, this possibility should be integrated into command and control system design at the earliest opportunity."

5.3.3 The "Software-First Machine"

"The software-first concept seems sufficiently promising to merit more detailed studies of its ramifications and alternatives, followed by exploratory or advanced development if appropriate."

5.3.4 USAF Hardware Laboratory

"The Air Force should not consider a production facility which would compete with private industry for Air Force production orders. An alternative worth further investigation involves creation of a USAF Hardware Laboratory along the lines of the Lincoln Laboratory, charged with research and development and limited production of prototype circuitry."

5.3.5 USAF Software Certification Authority (SCA)

"A more feasible alternative might involve establishment of a certification capability within an Air Force information processing technology staff organization. Rather than maintaining all certification responsibility, this portion of the organization could concentrate upon development of methodology, on-site assistance,and training of individuals from the responsible commands."

5.3.6 Desirable Objectives

1. "Provide more versatile, yet more economical and less manpower-intensive C&C operations for the 1980's."

2. "Reduce the typical C&C information-processing system development time from six to four years, and the resulting computer hardware age at IOC from three or four years to one or two years."

3. "Reduce significantly the danger that software errors could escalate crisis situations or degrade defenses at critical times."

4. "Provide survivable, high-capacity airborne computer capabilities allowing the functional equivalence of ground-based and airborne C&C operations."

5. "Provide combat-ready C&C information-processing systems which are far more reliable and responsive in their support of dynamic force management requirements."

6. Improve "requirements analysis and design techniques sufficient to save one man-day of effort per man-months."

7. Improve "software certification techniques sufficient to save one man-day of effort per man-month."

8. Increase "software transferability by 1%."

9. Increase "software productivity from 10 to 11 instructions per man-day."

10. Improve "computer system performance analysis sufficient to realize a 25-percent improvement in hardware system efficiency on only 25-percent of the Air Force's computers."

5.4 CORRELATION WITH APL RECOMMENDATIONS

The findings and recommendations from CCIP-85 correlate most closely with the following four APL recommendations: MP1, IP2, IP3, and MS1.

1. Analysis and Validation* of System Requirements MP1

   Direct that a comprehensive analysis and definition program be carried out on software (as well as hardware) elements of each new major Weapon System during the Program Validation Phase, prior to approval of Full Scale Development. The software definition should be carried down to the level of subprograms performing major functions.

   Cost estimates for the development and integration of each subprogram should be based on analysis, simulation, modeling, or construction of its principal parts, as called for by its respective newness or criticality.

2. Disciplined Programming IP2

   Require that the computer program development contractor apply a highly disciplined set of engineering practices to the detailed design and programming phases of development. This must involve a clear and disciplined set of standards covering program structure, size, control, interface, formal conventions on data base management, and the demonstration that the standards are enforced in practice.

3. System Integration and Test Capability IP3

   Require that an integration and test capability be provided as part of Full Scale Development of major Weapon System software, tailored to the specific needs of the program. This should be a software test-bed combining simulated elements and hardware (including operator consoles) to be used in progressive integration and test of system elements. It should provide real-time dynamic stimuli and responses under repeatable and off-nominal test conditions. The portion of this capability that is required for Operational Support and Maintenance should be specified to be transferable or capable of duplication.

*"Validation" is used in the context of Program Validation Phase, as opposed to validation/verification of coded computer programs.

4. Technical Staffing of Program Manager Organization MS1

Establish and implement a policy that Program Managers for major Weapon Systems be staffed with personnel experienced in systems engineering and software development and of sufficient stature and number to carry out essential management functions that cannot be delegated.

PROJECT PACER FLASH

6. PROJECT PACER FLASH

6.1 INTRODUCTION

Air Force Regulation 20-1, dated 4 December 1972, established Project Pacer Flash "for the purpose of conducting an in-depth study of long-range computer software support required for weapon system computers" and required the Air Force Logistics Command (AFLC) to chair the study group. The regulation also required the participation of the Strategic Air Command, the Tactical Air Command, the Material Air Command, the Air Defense Command, the Air Training Command, and the Air Force Systems Command.

"The task group was charged to:

(1) Determine the present and projected inventory of programmable computers installed in weapon systems;

(2) Review existing policy and procedures pertinent to support requirements;

(3) Develop changes to existing policy and procedures or develop new policy and procedures for weapon system computer support;

(4) Recommend an Air Force.position on management of software support, and;

(5) Publish and update new or changed policy and procedures in appropriate directives."

6.2 FINDINGS/CONCLUSIONS

6.2.1 Important Problems

When the study was initiated, several facts identified by the Air Staff advisors as being important and having a bearing on the problem were stated as follows:

1. "There is a lack of software development standards."

2. "The cost of contractor software support is significantly higher than in-house support."

3. "There is little or no transferability of software maintenance support (between or among weapon systems)."

4. "There is a lack of adequate (in-house) testing and validation of software programs."

5. "There is an apparent need for greater in-house capabilities, under a centralized management, to accomplish much of this software support."

6.2.2 Relevant Facts

Other factors bearing on the problem were stated to be factual information that should be understood and considered during the implementation of the study. The following statements comprise at least a partial list of relevant facts as presented in the Final Report:

1. It is expected that the complexity, scope, and cost of software support will necessarily increase because of the increase of weapon system complexity.

2. "Headquarters USAF has stated the need for expanding the long-range organic capability for software support."

3. It is essential for hardware and software support to be considered as an integral problem.

4. Computer "software requires configuration management."

5. In order to assure proper development and later support of software, expansion of Air Force policy and procedures must be accomplished to cover the entire life cycle of a Weapon System.

6. "Planning for software support (supportability) must begin during the conceptual phase of system design."

7. System/subsystem engineering and software expertise are both required for software support of Weapon Systems.

8. "An organic capability for software validation and verification is required."

9. Because computer software is the least visible and tangible of any subsystem, it follows that it is the least understood.

10. The software acquisiton process for avionics systems constitutes a major problem in the implementation of digital avionics.

11. The annual Air Force software expenditure is between $1 and $1.5 billion, which is approximately three times the computer hardware expenditure and is approximately 4 to 5% of the total Air Force budget.

12. "There is a scarcity of solid quantitative data to demonstrate the impact of software on operational performance or to provide perspective on R&D priorities."

6.2.3 Conclusions

1. Because software costs are continually rising, it is expected that by 1985 the cost ratio of software to hardware will be approximately 9 to 1. Also, the software costs, which in reality are people costs, are expected to exceed the $1 to $1.5 billion level of FY 72. Further, the Air Force expects that by expanding and improving its organic software capability a substantial saving will be attained.

2. "Training requirements will be substantial to bring existing software support personnel to fully qualified status and to augment the skills of the new hires even though the Air Force would attempt to recruit the best qualified personnel available."

3. Because it was very difficult to obtain an inventory of programmable computers, there apparently exists a need for the Air Force to strengthen the methods by which it controls and accounts for such devices.

4. "Air Force directives require revision or expansion (or new issue) to adequately cover the research, development, acquisition, operational and support phases of weapon system software considered by the PACER FLASH Study. Software must be accorded the same degree of management control presently accorded weapon system computer hardware."

5. "The Air Force should consider weapon system computer hardware and its related software as an integral problem — decisions regarding one should be made with full recognition of the other."

Included in the Final Report are approximately 35 additional conclusions dealing with a wide range of subject matter such as Air Force Specialty Codes, standardization, specified languages, documentation, management, procedures, reviews, automatic test system, cost analysis, and software as a deliverable item.

6.3 RECOMMENDATIONS OF PACER FLASH

1. "The Air Force should move in directions which will increase its organic capability for software support."

2. "Software support should be made the explicit responsibility of the weapon system manager at the AFLC AMA where he resides and for the aircraft for which he is responsible."

3. "Recognize common and unique requirements for software support in the ATE, OFP, and SIM areas as delineated in the report and physically and organizationally locate functions outside of AFLC (including AMAs) where these functions are most cost effective and responsive to user mission requirements. The system manager retains configuration management control."

4. "Revise or issue new directives as appropriate to address the requirement for a continuous inventory of programmable digital devices. Standardize the data item descriptions (DIDs)."

5. "Revise the Air Force Specialty Codes."

6. "Revise weapon system management directives to assure standardization of weapon system hardware/software during conceptual and design phases."

7. Standardize "languages for use in weapon system hardware/software."

8. "Revise configuration management policy and procedures to consider weapon system hardware/software as an integral problem under the system manager's responsibility and authority."

9. "Establish a central point of contact or office within AFSC to oversee the application of an Automatic Test System to a weapon system."

10. "Establish an automatic test equipment capability within AFSC Divisions to assure the application of the automatic .est system to weapon systems during DT&E and production."

11. 'Assure adequate development of weapon system software and documentation deliverable as contract items even though a reduction in the number of delivered weapon systems is the result."

6.4 CORRELATION WITH APL RECOMMENDATIONS

The findings and recommendations from Pacer Flash correlate most closely with the following three APL recommendations: MP3, IP1, and IP3.

1. Software as Contract Deliverable MP3

   Specify that major computer software involved in Weapon Systems development be designated Configuration Items (CI's) and deliverables during Full Scale Development. This would generally include computer programs and computer data for

   1. Operational Software,
   2. Development Support Software, and
   3. Test and Integration Software.

2. Software Development Support Tools and Facilities IP1

   Ensure that the Full Scale Development program includes provision for adequate modern support tools and facilities including such items as assemblers, compilers, editors, debugging aids, data base and library management systems, and associated operating systems. Require maximum use of existing proven tools and facilities. Provide that any of these tools and facilities that will be required bv the Operational Support (Maintenance) Agent for system maintenance be delivered in transferable form and also be capable of application to future Weapon System programs.

3. System Integration and Test Capability IP3

   Require that an integration and test c pability be provided as part of Full Scale Development of major Weapon System software, tailored to the specific needs of the program.

This should be a software test-bed combining simulated elements and hardware (including operator consoles) to be used in progressive integration and test of system elements. It should provide real-time dynamic stimuli and responses under repeatable and off-nominal test conditions. The portion of this capability that is required for Operational Support and Maintenance should be specified to be transferable or capable of duplication.

AUTOMATIC DATA PROCESSING COSTS IN THE DEFENSE DEPARTMENT

7. AUTOMATIC DATA PROCESSING COSTS IN THE DEFENSE DEPARTMENT

7.1 INTRODUCTION

Automatic Data Processing Costs in the Defense Department, Paper P-1046, was prepared by D. A. Fisner, Jr., of the Institute for Defense Analyses (IDA), in October 1974. As stated, the paper "attempts to provide substantiated estimates of the costs and cost trends of DoD computer software and other ADP activities, and the major components of those costs, on the thesis that a determination of present costs is the necessary first step in deciding on future DoD investments in ADP research and development." It was pointed out that specific cost items that were disproportionately high could lead to review and action in areas where urgent attention was required. It was further stated that "the estimates derived in this paper provide insufficient basis in themselves for making recommendations on the future course of DoD R&D on ADP, and no such recommendations are made here. The findings, however, provide an estimate of computer software costs to DoD and show the structure of those costs. This information is necessary for developing and evaluating DoD software research programs."

The CCIP-85 study, which was discussed in Section 5 of this Appendix, provides "the most extensive analysis available on computer software and ADP in DoD and CCIP-85 forms the basis for much of the current DoD thinking in this area." The estimates for the current study were compared with those made by CCIP-85, with some interesting results, which will be discussed later.

The estimates developed in this study are based on information from data sources that are readily accessible in DoD and other Government agencies. It was difficult to use all the data from all sources, however, because the information had been collected for other special purposes and was not organized in a readily usable manner. A major source of information for the paper was the Inventory of Automatic Data Processing Equipment in the United States Government, a GSA document "maintained under the requirements of the Brooks bill (Public Law 89-306, October 1965)." The Inventory was the "best single source of quantitative information on computers and ADP activities in DoD," including numbers, operational costs, and capital costs of General Management Classification (GMC) systems but only the numbers of Special Management Classification (SMC) systems. "The Inventory does not report on ADP systems used in weapons systems." The study, therefore, uses total Weapon System cost numbers "as reported in the DoD budget to estimate the cost of weapon system ADP systems." Other data from additional sources "included costs obtained from DoD procurement contracts and information from the Civil Service and the Military Services on the numbers and salaries of civilian and military ADP personnel in DoD."

"Pertinent data from various sources were identified, combined, and interpreted to determine a measure of the reported direct DoD ADP costs. Sufficient documented information on ADP activities and their costs was identified to determine a lower bound on costs and their probable component structure. Estimation was necessary to arrive at total software and ADP costs.

"Reported costs were partitioned among software, hardware, and other ADP activities, and an estimate of personnel burden was added. This provided a supportable but not necessarily correct figure for the size of reported DoD ADP costs and showed that software accounts for the major fraction of ADP cost to the Department of Defense.

"The costs thus arrived at were then used as a basis for estimating total annual reported and unreported DoD expenditures for ADP. Estimates were developed for the individual Services as well as for DoD as a whole.

"All assumptions made in developing the estimates are stated, and all calculations are explained."

7.2 FINDINGS/CONCLUSIONS

1. "Reliable information on most software and ADP costs in DoD is unavailable in a clearly identifiable form. The following data for FY 1973 were developed in this paper."

2. "Documented and identified annual ADP costs in DoD were $1.5 billion in FY 1973; this rises to $2.3 billion when an estimated cost of personnel burden is included."

3. "The estimated annual ADP costs in DoD are $2.9 to $3.6 billion for software and a total $6.2 to $8.3 billion when hardware and other ADP are included (approximately 30% to 50% of all electronics costs in DoD)."

4. "ADP costs in DoD are apportioned approximately as follows:

Software	45%
Computer Hardware	16%
Other ADP (includes computer operation, key punching, support and supplies)	38%"

5. "An estimated 70% of ADP costs in DoD are for personnel."

6. "The number of ADP man-years in DoD is divided almost equally between

   - Software systems analysis, design, and programming
   - Operation of ADP equipment (except key punching)
   - Services, support, and key punching."

7. "The annual costs of Air Force software and computer hardware estimated in this paper and in CCIP-85 are similar. This is surprising, because the CCIP-85 estimates are based entirely on analogy with industry and are not well documented."

8. "A comparison of DoD ADP costs reported for FY 1968 and FY 1973 shows that:

   a) Total ADP costs and total ADP personnel salary costs remained unchanged, while costs per system rose 4% to 5%.

   b) Total ADP contract service costs rose 54%, or 61% per system.

   c) Rental and capital costs for ADP equipment dropped 8%, which is 5% per system.

   d) The total of in-house ADP man-years dropped 10%.

   e) There was a shift from use of in-house personnel to contract services for system analysis/design, programming, and maintenance.

f) There was a shift from rental to purchase of ADP equipment.

g) The number of computer systems increased 28%, while the number of systems reporting ADP operational and capital costs and personnel activities declined."

As noted in the Introduction, the study did not make any formal recommendations. The principal findings, however, provide valuable information as background for the DoD Weapon Systems Software Management Study conducted by APL.

AIR FORCE LOGISTICS COMMAND OPERATION FLIGHT PROGRAM SUPPORT

8. A REPORT ON AIR FORCE LOGISTICS COMMAND OPERATION FLIGHT PROGRAM SUPPORT

8.1 INTRODUCTION

In July 1974, System Development Corporation, under contract to Warner Robbins Air Logistics Center (ALC), initiated a study of the support methodology used by various other Air Logistics Centers within the Air Force Logistics Command (AFLC). The study was directed to selected transitioned Weapon Systems and their respective Operational Flight Programs (OFP's).

During the past several years there has been a definite trend toward the use of digital elements in aeronautical systems. "The software associated with programmable devices, typical to integrated avionic systems, has emerged as one of the key elements affecting avionics and weapon system performance. It has become evident that the Air Force requires a capability to effectively manage contractor developments in the software area which is as effective as the Air Force's capability to manage hardware procurements." Because the Air Force in-house capability to support in-depth software changes is generally limited, it has been necessary to retain the development contractor to provide software support even after transition from Air Force Systems Command (AFSC) to AFLC.

The stated objectives for the final report included the following:

1. "Preparation of a summary of the final report which could be used by AFLC management personnel."

2. "Creation and documentation of the data which was collected by SDC from the various ALCs. This data was divided into six major sections:

   (a) Characteristics of the OFPs and their associated weapon systems.

   (b) Current support posture of the OFPs.

   (c) Personnel required in the support of the OFPs.

   (d) Documentation required in the support of the OFPs.

   (e) Configuration management required in support of the OFPs.

   (f) Testing in support of the OFPs."

3. "Performance and preparation of the analysis of the collected data."

4. "Preparation of the conclusions and recommendations derived by SDC in its analysis."

5. "Preparation of the extraneous data collected which proved germane to the report, but did not fit into previously defined sections of the report."

8.2 FINDINGS/CONCLUSIONS

1. "Based upon the survey performed for the aircraft OFPs listed in the Statement of Work, it was found that Configuration Management practices and procedures are not performed in a standard manner. In some cases, a baseline configuration for OFP software did not exist, while in others, maintenance of the baseline was provided through very informal techniques. Because of the manner of organizing Configuration Control Boards, some of the technical personnel in charge of software maintenance tend to avoid formal management procedures."

2. "The Technical Order system for control of software updates was found in many applications. This system applies, at best, to maintaining a library of program versions and does not address the subject of program documentation. Standard software specification processes are understood by the technical personnel responsible for software maintenance and most groups informally maintain operational software specifications. In order to adequately maintain documentation, the documentation must be initially delivered by the contractor in charge of OFP software and procedures for documentation update followed."

3. "In most cases, support software was not provided to AFLC for the OFPs at transition. Obviously, organic support of OFP software is not possible without this important support software. In the future, any procurement of avionics software must include provisions for definition of the support software as a CPCI for delivery. In those cases below where organic support capability is recommended, then an obvious requirement for procurement of this software exists."

4. "In no case was a condition found that standardization of programming languages, or the rewrite of a program from one

system to another, was warranted. In the first case, the computers utilized for avionics systems are generally memory limited. Some have quite limited instruction sets. These bounds have forced a preference for Assembly Language level programming in order to reduce program size. While it is felt that future systems should tend to a standardization of computers and associated support software, the nature of present systems does not lend itself to standardization."

5. "The possibility of rewriting a computer program from one computer to another is not feasible, again, because of the extreme differences between machines. However, as was found in several navigation systems, the modification of a computer program is feasible (and is done) where the same navigation system is used in a different aircraft."

6. "The subject of Validation, Verification, and Certification was found to present significant difficulties in discussion in OFPs. This difficulty arises because of the wide diversities in defining V,V&C. It was found more convenient to discuss the various levels of OFP testing followed for the evaluation of OFPs."

7. "Only in the area of nuclear safety will it be necessary for a truly independent third party testing process to be carried out. Also, this independent testing will be necessary only if the new nuclear safety regulation, AFR-122-9, is found to apply to manned aircraft. If the regulation applies, it will be necessary to provide this form of testing to SRAM."

8. "In the majority of other cases, test plans and procedures are currently being prepared and adhered to in the evaluation of software. It must be assured that this practice is being followed in all cases."

9. "A final conclusion which can be made from the data gathered in this study is that personnel within AFLC responsible for software support of avionic computers have a difficult task to perform."

3.3 RECOMMENDATIONS OF THE REPORT ON AIR FORCE LOGISTICS COMMAND OPERATION FLIGHT PROGRAM SUPPORT

1. "Baseline software configurations should be established and proven techniques for the management of this configuration must be adhered to."

2. "All future avionic system buys should include provision for delivery of support software. For those systems specifically studied in the scope of this project, the support software should be procured, after the fact, in the majority of cases."

3. "The Technical Order system with respect to software control should be phased out and replaced with the proven specification system, defined in MIL-STD 800-14 and amplified in AFM 800-XX."

4. "With respect to future support of the OFPs listed in this study, it is recommended that the combination of Weapon System Manager and Functional Area organizational structure be further defined and implemented to manage all OFP support activities. A part of this recommendation includes the dedication of a large scale general purpose digital computer to OFP support. This computer is required by the SRAM support lab and is proposed to be located at Oklahoma City, ALC."

8.4 CORRELATION WITH APL RECOMMENDATIONS

The findings and recommendations from this AFLC OFP report correlate most closely with the following two APL recommendations: IP1 and MS3.

1. Software Development Support Tools and Facilities IP1

   Ensure that the Full Scale Development program includes provision for adequate modern support tools and facilities, including such items as assemblers, compilers, editors, debugging aids, data base and library management systems, and associated operating systems. Require maximum use of existing proven tools and facilities. Provide that any of these tools and facilities that will be required by the Operational Support (Maintenance) Agent for system maintenance be delivered in transferable form and also be capable of application to future Weapon System programs.

2. Software Operational Support Agent MS3

   Require that the Software Operational Support (Maintenance) Agent be identified and consulted during the Program Validation Phase to support the Program Manager in providing for maintenance support requirements. Require that the agent be included at the beginning of and throughout Full Scale Development to plan for system integration, testing, and transfer from developmental to operational status.

AERONAUTICAL SYSTEMS SOFTWARE WORKSHOP

9. PROCEEDINGS OF THE AERONAUTICAL SYSTEMS SOFTWARE WORKSHOP

9.1 INTRODUCTION

Because recent studies such as Project Pacer Flash had emphasized and recommended the need to exchange communications at all levels, the Aeronautical Systems Software Workshop was held on 2 to 4 April 1974 to provide an opportunity to exchange information on avionics systems software. Software, because of costs and lack of visibility as to progress etc., had received a great deal of management attention and concern. The attendees at the Workshop included personnel from the Air Staff, Air Force Systems Command (AFSC), Air Force Logistics Command (AFLC), the User Commands, and other Government agencies. Members of industry and DoD presented papers on how they had successfully met and coped with many problems involved in the avionics systems software acquisition process.

In his opening remarks to the Workshop, Maj. Gen. Douglas T. Nelson set the stage for the presentations and discussions by emphasizing the major importance of software to achieve flexibility for our modern airborne Weapon Systems. Further, he pointed out that "accompanying our digital trend with its flexibility has been the increased involvement with the attendant software: all the way from design, development, through test and transition to the Air Force Logistics Command and operation by the using commands. Our experience has pointed out several problem areas in software which will be discussed during the Workshop." He also discussed other areas that cause problems in the software acquisition process: increasing cost for the entire life cycle of software, maintenance requirements, crew trainer simulators, the B-1 program with its many required system interfaces, and others. Another statement by Maj. Gen. Nelson summarized in a few words the total objective of managing the software acquisition process: "If any part of our development job ever called for skillful systems management, this task of bringing the whole weapon system together with software certainly does!!!"

(The following four paragraphs are paraphrased from the introductory part of the Proceedings.)

Gen. Nelson was followed by Col. G. Fernandez, Assistant for Processor and Software Planning, DCS/Development Plans, Headquarters AFSC, who presented a brief Program Overview. Col. Leo Danielian, the "spark plug" of Pacer Flash, followed next with his view from the Air Staff, Plans and Operations st 'point.

Next, three papers were presented dealing with the status of the three major categories of Aeronautical Systems Software: "Operational Flight Software," "Automatic Test Equipment Software," and "Crew Training Simulator Software," presented by Lt. Col. Edward S. Hinton, Richard C. Behymer, and Philip S. Babel, respectively.

Mr. J. D. Schmidt, Headquarters AFLC, presented his feelings with regard to the AFLC and user requirements for software maintenance. This was followed by Lt. Col. J. G. Daye's presentation of a description and status of the new Air Force manual, AFM800-XX (not yet released) on Weapon System Computer Resources Acquisition, Use and Maintenance.

On 4 April Col. Fernandez presented his summary of current and planned Air Force actions to solve aeronautical systems software problems. This presentation was followed by a panel discussion moderated by Lt. Col. Manley. In this panel discussion, chairmen of the earlier Contributed Paper Sessions summarized their individual sessions. These summaries are not explicitly presented in the Proceedings but their essence is included in Lt. Col. Manley's Overview of Contributed Papers. Col. Fernandez then closed the Workshop with some remarks on future plans.

It is important to note that there was no formal set of recommendations prepared from the Workshop. The individual papers address a wide range of software problems, and many solutions were described. Certain papers do list specific recommendations, however, and these have led to direct support of the formal recommendations prepared by APL for the DoD Software Management Study.

9.2 SUBJECTS COVERED

The subject areas covered at the Workshop included the major APL areas of (a) Management Policy, (b) Acquisition Planning, (c) Systems Engineering, (d) Implementation Procedures, (e) Program Management Support, (f) Acquisition Management Standards, and (g) Development of Tools and Techniques.

Additional specific subjects included Simulation, Theory (R&D), Hardware, Configuration Control, Higher-Order Languages (HOL's), Automatic Test Equipment, Compiler Development, Support Software, In-House Support, Software Maintenance, and Testing, Verification, and Validation.

9.3 RELEVANCE TO APL RECOMMENDATIONS

Each one of the 17 APL recommendations is supported by one or more of the individual Aeronautical Systems Workshop papers. It is informative, therefore, to give selected quotations from both sources for the 17 major areas.

9.3.1 Analysis and Validation* of System Requirements MP1

APL

Recommendation

Direct that a comprehensive analysis and definition program be carried out on software (as well as hardware) elements of each new major Weapon System during the Program Validation Phase, prior to approval of Full Scale Development. The software definition should be carried down to the level of subprograms performing major functions.

Cost estimates for the development and integration of each subprogram should be based on analysis, simulation, modeling, or construction of its principal parts, as called for by their respective newness or criticality.

Remarks

. . . Thus, the implementation of this recommendation requires the accomplishment of limited preliminary design during the Program Validation Phase, rather than as the first step of Full Scale Development.

Aeronautical Workshop

From E. R. Mangold, "Software Management and Visibility," p. 143

From a management standpoint it is essential that the successive steps in the development process be restricted until the preliminary design is complete. Experience has proven that resources expended in preliminary design early in the development process have greatly reduced down-stream testing and maintenance costs. The designers should be forced to complete this step even if it is necessarily guesswork requiring subsequent iteration. The preliminary design requires relatively little resource since it is best done by a few senior designers and programmers, but it is a difficult function because it is necessarily done early when knowledge of the system is minimal.

*"Validation" is used in the context of Program Validation Phase, as opposed to validation/verification of coded computer programs.

9.3.2 Software Visibility in Weapon System Acquisition MP2

APL

Recommendation

Increase the visibility and understanding of major software components of Weapon Systems by putting them on a par with hardware components. This is to be done in terms of configuration control items, DSARC reviews, design reviews, and other aspects of acquisition management.

Remarks

The Joint Logistics Commanders' SRWG is proposing a revision to MIL-STD-881 to call out software subsystems at the proper level in the work breakdown structure, as well as other actions to give software more visibility.

Aeronautical Workshop

From W. L. Trainor, "Trends in Avionic Software — Problems and Solutions," p. 106

The key concept underlying all of these tools is the elevation of software to the status of a Contract Item or Subsystem; a status which previously has been reserved only for hardware.

9.3.3 Software as Contract Deliverable MP3

APL

Recommendation

Specify that major computer software involved in Weapon Systems development be designated Configuration Items (CI's) and deliverables during Full Scale Development. This would generally include computer programs and computer data for

1. Operational Software,
2. Development Support Software, and
3. Test and Integration Software.

Implementation

Provide clear guidelines for designating appropriate computer system resources (computer programs and computer data) as CI's in the Program Management Directives and manuals. Call for scheduled delivery, like hardware items. Specify support and test and integration software as separate deliverables.

Aeronautical Workshop

From W. L. Trainor, "Trends in Avionic Software — Problems and Solutions," p. 106

The key concept underlying all of these tools is the elevation of software to the status of a Contract Item or subsystem; a status which previously has been reserved only for hardware. The time has come to single out software as major deliverables under contract terms, and no longer "drag along" software as some ill-defined and ancillary part of a hardware item.

9.3.4 Milestoned Development Plan API

API

Recommendation

Define the requirements for milestones in the Full Scale Development phase to ensure the proper sequence of analysis, design, implementation, integration, test, and review processes. Also, define criteria that will be used to demonstrate that each milestone has been achieved.

Implementation

Amplify the definition of requirements for Preliminary Design Review (PDR) and Critical Design Review (CDR) to specify the items of analysis, design, implementation, integration, and testing to be completed. Develop an updated version of the milestone definition of SSD 61-47B or its equivalent. (Note that Milestone 1 of SSD 61-47B should precede Full Scale Development.) Incorporate these in the Program Management Plan and specify that the milestone provisions be written into development contracts.

Aeronautical Workshop

From Lt. Col. E. S. Hinton, "Operational Flight Software," p. 20

In an attempt to increase management visibility during OFP development, a method of documentation is being used on the B-1 program whereby "milestone" events are identified and tracked in detail. These events are defined to be the publication and release of specified software documents which include the system specifications, module descriptions, interface documents, user manuals, etc. Milestones have previously been used to define events, which once passed, are complete and do not need to be redone. This has proved very successful for hardware oriented programs. On the other hand, software development is by nature very iterative and "milestoning" in this area is not as clear-cut. However, the emphasis on providing management visibility is proving to be invaluable to the program.

9.3.5 Computer System Resource Development Plan AP2

APL

Recommendation

Ensure provision of a detailed Computer System Resource Development Plan as part of the bid package on Full Scale Development contracts The plan should cover all aspects of the contractor's approach to organization, design, test, management, documentation, and other aspects of the program.

Problems Addressed

In order to ensure that the development of a major software subsystem is well organized and managed, and all requirements are properly understood and defined, it is essential to have a detailed development plan prepared and evaluated prior to starting Full Scale Development.

The development plan should include a detailed statement of the contractor's engineering and management approaches, and hence can serve as a basis for selecting a contractor with the requisite understanding, experience, and facilities.

Implementation

Require that the Program Manager prepare a set of development requirements to be included in the Request for Proposal (RFP). Specify those aspects that are directed by the Government. Specify the nature and scope of description required in the contractor's Computer System Resource Development Plan.

Aeronautical Workshop

From Lt. Col. J. G. Daye, "AFM 800-XX Computer Resources Acquisition Use and Maintenance," p. 52

A new deliverable to be identified in the manual is the computer program development plan which will be submitted with the contractor's proposal. The objectives of the development plan are to provide the program office with the necessary information to assure the PM that the contractor knows what he is doing, and provide the necessary muscle to monitor the progress and force any corrective actions.

9.3.6 System Engineering of Computer Systems SE1

APL

Recommendation

For systems involving several distinct functions, require that the system be divided into functional segments in accordance with the operational requirements. Require during the Program Validation Phase that tradeoff analyses be performed for hardware versus software (i.e., hardwired versus programmable functions) and for different computer system architectures.

Problems Addressed

The lack of application of systems engineering methodology to computer system design is at the root of a number of critical problems in the development of major Weapon Systems. It results in inefficient processing architecture, lack of hardware/software tradeoffs, and overcentralization, leading to overly complex requirements and hence large, cumbersome, and costly software programs.

Aeronautical Workshop

From Lt. Col. E. S. Hinton, "Operational Flight Software," p. 21

Good performance from OFP can only be assured when system design has been appropriately considered and tradeoffs between hardware and software have been made intelligently. Systems engineering plays an important role in the early phases of software development and the basis for decisions that must be made is an in-depth analysis and system design capability.

From Col. C. H. Allen, "ASD Involvement in Software," p. 166

The McAir software programming philosophy required that all software tasks generic to a given subsystem be accomplished by that subsystem and that mission oriented tasks should be accomplished by the central computer. This requirement appears to have established a clear dividing line on the software programming responsibility. The programming of the central computer has been kept less complex; . . .

9.3.7 Provisions for Growth in System Requirements SE2

APL

Recommendation

Provide for growth and change in requirements on Weapon System computer software by identifying parameters that are uncertain or are likely to change in the future and, where possible, specify the probable limits on such changes. Also identify novel environments and use of new techniques. Require that computer systems be sized to provide for uncertainties and requirement growth.

Problems Addressed

The nature of Weapon Systems is such that the inevitable growth and change in the enemy threat, as well as advances in sensor and weapon technology, result in corresponding growth and change in system requirements throughout the life of the system. While, in principle, changes in software should be less expensive than those in hardware, such changes can actually be extremely costly unless provisions for growth and change have been made in the initial design. Also, opportunities for designing-to-cost are frozen out unless provision is made for growth.

Aeronautical Workshop

From Lt. Col. E. S. Hinton, "Operational Flight Software," pp. 13 and 17

Software which performs the required functions is most useful when it is sufficiently flexible or changeable so that quick modifications can meet urgent mission requirements.

Increasing program size and cycle time are frequently problems in OFP development. They can result from poor requirements or they can be caused by optimism on the part of the development team. Air Force naivete in estimating program size has, in certain instances, allowed software contractors to propose computer memory requirements which were unrealistic, as was later determined.

From Col. C. H. Allen, "ASD Involvement in Software," p. 157

Technology is going to continue. Digital systems are going to be smaller, computer speed is going to become faster, computer memory is going to be cheaper and more readily available, and all the other good things which go along with technology advancement. Needless to say, this will allow for more capabilities to be implemented, which will, in turn, create new problems to be solved.

9.3.8 Systems Engineering of Computer Software SE3

APL

Recommendation

Specify the use of modular software architecture and an orderly, phased design approach for developing major computer programs that defines the higher levels of the program and then progresses to design and test successively lower levels. The latter approach is often referred to as "top-down" design. It involves the formal definition of a hierarchy of program elements and restrictions concerning lateral communications.

Problems Addressed

The lack of application of systems engineering methods to the design of software has led to systems that are nonmodular, lacking well-established interfaces, and difficult to test. The design approach has often been undisciplined, with implementation started before the overall structure has been defined. This results in incompatibilities and errors that are discovered late in the test process, with serious impact on schedules and costs. Lack of modularity results in complex interfaces and difficulties in accommodating to changes in requirements.

Aeronautical Workshop

From J. D. Schiff, "An Overview of the Software Life Cycle Process," p. 114

Top Down Development

Top Down Development is a development method which gives order to the implementation of the software system. From specifications and interfaces, the complete package is constructed beginning with the highest levels of control. The effect of this approach is two-fold. First, the system integration effort occurs simultaneously with the development; and second, an increasingly capable operational system is in use during development. The benefits associated with employing the top down development approach are:

Earlier detection of design problems

Orderly and comprehensive test development

Elimination of separate systems integration

Easier to isolate problems

Minimizes impact due to changes

9.3.9 Software Development Support Tools and Facilities IP1

APL

Recommendation

Ensure that the Full Scale Development program includes provision for adequate modern support tools and facilities, including such items as assemblers, compilers, editors, debugging aids, data base and library management systems, and associated operating systems. Require maximum use of existing proven tools and facilities. Provide that any of these that will be required by the Operational Support (Maintenance) Agent for system maintenance be delivered in transferable form and also be capable of application to future Weapon System programs.

Problems Addressed

The development of software requires a major investment in support tools and facilities. If they are not available from previous programs and are not provided for in the development plan, a major schedule slippage and cost overrun can result. If they are not designed to be transferable to the Operational Support Agent, as required for system maintenance support, additional costs will be incurred during the maintenance phase. Inadequate support tools lead to excessive test times and late detection of errors.

Aeronautical Workshop

From Lt. Col. E. S. Hinton, "Operational Flight Software," p. 18

Support software, such as assemblers, compilers, link-editors, simulations are all required if manipulation and modification of the OFP is to be done. Some of these software packages could be written in-house if they were not provided by the acquisition contract. However, the most cost-effective way to get them would probably be to call for their delivery under the initial contract since they had to have been available for the development effort by the contractor.

From W. L. Trainor, "Trends in Avionic Software — Problems and Solutions," p. 102

But once again, the same 'reinvent the wheel" philosophy that was noted to apply to OFPs is also applied to the support software. Each aircraft procurement has resulted in the development of new support software to accommodate new flight computers on new host computers. Not only is the redevelopment cost outrageous, but when all the smoke clears, Uncle Sam owns none of these support packages — which, incidentally, are needed to maintain the new OFPs during the many years of operational use. This was true on numerous projects. Support software was considered vendor proprietary and the supplier was "wired-in" for all future maintenance work: a costly situation, to say the least.

9.3.10 Disciplined Programming IP2

APL

Recommendation

Require that the computer program development contractor apply a highly disciplined set of engineering practices to the detailed design and programming phases of development. This must involve a clear and disciplined set of standards covering program structure, size, control, interface, formal conventions on data base management, and the demonstration that the standards are enforced in practice.

Implementation

The Request for Proposal (RFP) should call for a description of the contractor's design and coding manuals and his approach to programming discipline in the Computer System Resource Development Plan. Formal and well-established procedures that have been demonstrated on prior programs should be an important element in the contractor selection process. The contract should specify that the proposed procedures be used.

Aeronautical Workshop

From Col. C. H. Allen, "ASD Involvement in Software," p. 162

> Does he demonstrate good clear modular programming, and does he specify common programming techniques for all module engineers so that the operational flight program is readily supportable in-house or by other contractors?

From W. L. Trainor, "Trends in Avionic Software -- Problems and Solutions," p. 104

> The use of well-designed production standards would serve as a mold to direct the programming personnel's thoughts and actions toward a logical and common end product — "quality" software. Particular areas which such standards would typically address are: (a) Coding conventions; such as, use of indentations, spaces, blank lines, etc., to increase readability; (b) Documentation conventions; such as, use of "comments" within program listing to improve "ease of comprehension"; (c) Labeling and naming conventions and restrictions to produce common and consistent terminology within the program; (d) Instruction-use

conventions/restrictions; such as, use of "GO TOs", if ever; (e) Conventions for parameterization, reuse of modules, etc.; (f) Conventions for assigning attributes to data/constant types; (g) Input/output conventions and restrictions; (h) Etc.

9.3.11 System Integration and Test Capability IP3

APL

Recommendation

Require that an integration and test capability be provided as part of Full Scale Development of major Weapon System software, tailored to the specific needs of the program. This should be a software test-bed combining simulated elements and hardware (including operator consoles) to be used in progressive integration and test of system elements. It should provide real-time dynamic stimuli and responses under repeatable and off-nominal test conditions. The portion of this capability that is required for Operational Support and Maintenance should be specified to be transferable or capable of duplication.

Implementation

Define the provision of an integration and test capability as a requirement in the Request for Proposal (RFP) and in the Computer System Resource Development Plan. Specify that portion of the simulation software as a contract deliverable, with formal documentation, as will be required for system operational support and maintenance. Provide O&M funds to the contractor for support of maintenance features. Constrain sophistication to avoid overcomplication, especially at the contractor facility. Make provisions for Integration and Test Facility planning in Acquisition Management regulations, and subject such planning to design review procedure. Consider training requirements for test facilities.

Aeronautical Workshop

From A. E. Patterson, "Sacramento Air Logistics Center, F-111 Avionics Integration Support Facility," p. 323

This capability is centered around the development and implementation of the F-111 Avionics Integration Support Facility (AISF). This facility provides unique laboratory support for OFP development/verification and avionics system integration.

To increase the AISF effectiveness, a program has recently been initiated to add a dynamic simulation area. This area shall provide a capability to conduct complete dynamic testing of all modes and functions in the OFPs. . . . This will mean that approximately 87 percent of the OFP performance can be checked in the laboratory as opposed to 60 percent at the present time and will reduce the OFP flight test requirements by approximately 50 percent.

From Col. C. H. Allen, "ASD Involvement in Software," p. 162

Specify in software contract a dynamic simulation capability sufficient to provide good software programming verification and validation before the flight test phase begins. Flight test is increasingly expensive.

If possible, specify a software integration and test, and dynamic simulation facility be built on Air Force property intended for life cycle support after development completion by contractor. This facility would be used by both Air Force and contractor, then remains in Air Force hands for software support after the development cycle.

9.3.12 Technical Staffing of Program Manager Organization MS1

APL

Recommendation

Establish and implement a policy that Program Managers for major Weapon Systems be scaffed with personnel experienced in systems engineering and software development and of sufficient stature and number to carry out essential management functions that cannot be delegated.

Implementation

Provide for high level review (e.g., DSARC I and II) of Program Manager staffing at the start of Program Validation and the Full Scale Development Phases of major Weapon System development programs. Provide means for temporary assignment of engineers from service laboratories and support activities to fill key staff positions. Provide career incentives to attract competent engineers from within and from outside the Government into both military and civilian positions. Establish policies that assure adequate grade levels for Civil Service jobs in this area.

Aeronautical Workshop

From M. R. Davis, "Visibility and Responsibility in Areonautical Systems Software," pp. 85 and 92

As in industry, senior people tend to migrate into management positions; in the Air Force you expect to find senior people in management slots and, in general, they are not going to be current in the state of the software art. These ROC's must be looked at by people who are up to speed in contemporary software business. So, somehow these senior people who sign off on technological feasibility need to be backed by people who can help them attain the awareness they need. They need someone who can assess the magnitude of the software problems relative to the state of the art, and who knows what is being done in research, and the prognosis for that research. . . .

What kinds of incentives are needed to induce the right kinds of people to come on board (both civilian and military)?

Is pooling of manpower resources a sensible interim solution until more people can be acquired?

9.3.13 Systems Engineering Agent MS2

APL

Recommendation

Establish a policy that, for major new Weapon System programs, the Program Manager engage a Systems Engineering Agent to assist in problems arising in the translation of system requirements into detailed hardware and computer system design requirements. The agent, whether Government or contractor, should be highly experienced in system operational requirements, special purpose system hardware, and computer system software and hardware.

Problems Addressed

Although the Program Manager should have on his immediate staff systems engineers who are knowledgeable about software, manpower limitations often restrict the staff to a skeleton organization. Without other direct support, the Program Manager cannot adequately fulfill his responsibilities for carrying out the extensive planning and monitoring associated with a major new Weapon System. This can result in insufficient definition of requirements, limited requirements analysis, unrealistic schedule and cost estimates, and inadequate configuration management.

Aeronautical Workshop

From Lt. Col. E. S. Hinton, "Operational Flight Software," p. 20

> Accompanying the contracted effort, is the utilization of "in-house" (civil service and military) personnel to monitor the software development in depth. This implies "hands-on" review of the software with an associated buildup of familiarity during development. In order to support this concept, a buildup of personnel with software engineering expertise is in process.

From Col. C. H. Allen, "ASD Involvement in Software," p. 158

> Special emphasis was given to avionics system engineering in the reorganization of the Directorate of Avionics Engineering at ASD. The recognized concepts to improve our software situation cannot be implemented unless a strong system engineering capability exists. Personnel are needed who are knowledgeable in several technical disciplines to a reasonable depth, who understand the

trade off between system/subsystem design alternatives and trade offs between requirements, capability, and cost. Good system engineers can only be developed through practical experience.

9.3.14 Software Operational Support Agent MS3

APL

Recommendation

Require that the Software Operational Support (Maintenance) Agent be identified and consulted during the Program Validation Phase to support the Program Manager in providing for maintenance support requirements. Require that the agent be included at the beginning of and throughout Full Scale Development to plan for system integration, testing, and transfer from development to operational status.

Problems Addressed

The integration of operational support requirements and the transition from production into operational use are high on the list of major problems in Weapon Systems acquisition. The lack of transferability of software, the lack of provisions for maintenance, and the cost of changes resulting from these inadequacies have been cited in many previous software studies as important problems needing solution.

Implementation

Amplify those parts of the Program Management Plan and the Program Management Directive dealing with the early participation of the Using and Supporting Commands to include the identification of an Operational Support Agent. Provide means for applying O&M funds to support contractor activity directed toward providing maintenance capabilities and documentation.

Aeronautical Workshop

From R. Fischer, "F-111 AGE Software, Generation, Maintenance and Transition to AFLC," pp. 390-391

One of the problems experienced has been lack of the assignment of individuals at each of the organizations with transfer responsibility.

Recommendation

Early in any program, the contractor, ASD, and AFLC should appoint specific individuals to be responsible for software transfer. These people should plan the eventual transfer and be responsible for its implementation.

In addition, consideration should be given to assigning government personnel, experienced in software, from the eventual user organization to the contractors facility during software development. Familiarity of those government personnel with the development process will eliminate the "credibility gap" on the scope of software transfer tasks.

From R. J. Schlight, "A Functional Approach to Software Management," p. 132

User/Developer Interface. The interface between the user and the software system developer is critical. It is vital (1) that he expresses his requirements clearly, (2) that he is aware of the limited capabilities of any software system, and (3) that the user understands what he is going to receive. The user should be involved in the software development cycle, should view the system in operational stages, and should make meaningful contributions which will ensure the responsiveness of the finished product.

9.3.15 Standard Criteria for Weapon System Computer Resources Acquisition Management AM1

APL

Recommendation

Establish a common set of requirements and criteria to be applied in the acquisition and support of Weapon System computer resources by all services.

Problems Addressed

Many of the preceding recommendations have proposed policies with regard to various aspects of system acquisition management. Their implementation requires the establishment of one or more top-level documents that would constitute official guidance to Program Managers and contracting officers. Current MIL-STDS on this subject are not adequate and are primarily hardware oriented. Variation in terminology is another problem that must be addressed to reduce confusion and the misinterpretation of existing guidelines.

Implementation

Derive a tri-service document covering the procedures to be used in the acquisition and support of Weapon System computer resources, using current service regulations and manuals as a basis. . . . Use a common terminology along the lines recommended by the Joint Logistics Commanders' Software Reliability Work Group.

Aeronautical Workshop

From R. W. Wolverton, "Paradoxes in Management: Software Standards and Procedures," p. 205

Our premise is that software standards and procedures have proliferated in number over the past several years and are, despite well-intentioned actions on the part of the government, causing more problems than they are solving as viewed by industry. The problem is characterized by a feeling of being out of touch with the informational activities of others, of a too rapid growth of what there is to know. The results of these separate acts are not converging at present. We will examine government rules and regulations which influence the contractor's technical approach, his software development procedures, and his product control in view of the dilemma presented by proliferating — and often contradictory or silent — government procurement procedures.

9.3.16 Software Acquisition Guides AM2

APL

Recommendation

Prepare a series of handbooks or guides covering important aspects of software acquisition, to help Program Managers and their staffs to define, review, and evaluate requirements, procedures, proposals, and designs during pre-contract and contract management. These would include such items as

Life Cycle Plan
System Requirements Review
RFP Preparation and Review
Computer Resource Development Plan Review
Preliminary and Critical Design Review
Documentation Standard Selection
Support Facility Plan Evaluation
QA Plan Evaluation

Problems Addressed

The great variation in the requirements and structure of Weapon Systems, differences between new and evolutionary systems, different methods of contracting, and organization of the sponsoring agency all require a large degree of flexibility in the application of management standards and procedures. However, the abstract nature of software and the relatively underdeveloped systems engineering methodology make it very difficult for Program Managers and their limited staffs to apply the necessary judgment in the absence of an organized body of knowledge to guide them.

Implementation

Coordinate current service efforts or assemble a tri-service committee with government and industry representation, under the sponsorship of OSD, to prepare suitable handbooks. Issue drafts for interim guidance and to obtain feedback from experience. Allocate special funds to participating service agencies.

Aeronautical Workshop

From R. Fischer, "F-111 AGE Software Generation, Maintenance and Transition to AFLC," p. 392

In an effort to define documentation the Air Force has produced several significant documents during the past 18 months. Three of these documents are listed as follows:

a. AFLC Reg. 66-37, "Management of Automated Test Stations"
b. AFLC Reg. 66-27, "Automated Support of Numerical Control and ATE Software"
c. SAALC/MMD, "ATE Acquisition Planning Guide"

These documents are excellent for use in identifying and defining responsibilities and documentation and provide a good basis for future programs.

Recommendation

A document, which specifically addresses only the transfer problem, should be prepared. This document should contain a specific plan to be adhered to in the transfer of any new program. It should include a step-by-step transfer plan for any new program, a list of all documentation requirements and a description of the responsibilities assigned to the contractor and the government agencies.

9.3.17 Software Test Tools TT1

APL

Recommendation

Support development of improved software test and validation tools to reduce the cost and time involved in software verification. These should include automated tools to identify and exercise all branches, to detect and isolate design faults, and to categorize error sources.

Problems Addressed

Test and validation has been the most time-consuming phase of software development. This has been true not only because of the numerous errors introduced by poor design methodology but also because of the effort required to design test drivers for individual portions of the program. In addition, manual testing of the full range of possible input conditions (in order to exercise all portions of the program) is extremely time consuming. Finally, the generation and running of test programs are subject to human error, which further adds to the validation time.

Implementation

Support ongoing service programs in development of automated test and validation tools. Fund the conversion of selected tools to the high level languages used in Weapon Systems (e.g., CMS-2, JOVIAL) and provide them to system contractors and Operational Support Activities as soon as economically practicable. Invite innovative proposals for new work. Support R&D efforts in software portability to aid in the application of tools to different systems.

Aeronautical Workshop

From R. E. Wattenburg, "Independent Test and Evaluation," p. 337

To aid in the code evaluation task, a series of automated tools have been developed. These tools primarily work with programs in a static sense, analyzing code instead of executing it. These tools are quite sophisticated and are almost all fully automated. They include automatic equation and flowchart generators (from object code), comparitors, editors, path analyzers, etc. The vast majority of programming classes of errors (data declarations, symbol duplication, improper register usage, etc.) can be detected easily and early using this type of tool.

From D. A. Ziener, "An Approach to Verification and Validation of Operational Software," pp. 350-351

> The ACS [Automated Checkout System] is a series of programs written for a large commercial computer. When a program for the airborne computer is operated on by the ACS, it performs a symbolic simulation and generates symbolic equations (A = B + C x D) that are performed by the airborne computer program. The ACS performs many checks for program errors, indicates all areas where analysis is required, and furnishes the program with the information required for analysis in optimum form. The equations generated by the ACS are manually checked against the input specification.

From J. D. Baum and J. B. Di Stefano, "Avionics In-Flight System/Software Test Tool — Anomaly Trace," pp. 356-357

> The idea of the anomaly trace tool was that of a dynamic instrumentation monitor. . . . it would not operate continuously but would lie dormant with only part of its code being exercised under normal conditions. Once an anomaly was detected, however, the anomaly trace tool would become active and seize full control of the sequencing of the real-time computer. The anomaly monitor would not only dump system data to be used in later analysis, but it would actively try to detect the occurrence of anomalies and to restore the system to normal status once one was observed.
>
> It is to be noted here that the anomaly trace system was developed entirely in the footstep of the existing system. No new equipment was added. The only change made to the system was the addition of a few hundred words of computer code.
>
> Program Tracing Error — The occurrence of this anomaly indicates that programs were executed out of their proper sequence.

MONTEREY SYMPOSIUM ON THE HIGH COST OF SOFTWARE

10. PROCEEDINGS OF A SYMPOSIUM ON THE HIGH COST OF SOFTWARE HELD AT THE NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA, ON SEPTEMBER 17-19, 1973

10.1 INTRODUCTION

The Monterey Symposium on the High Cost of Software was held in September 1973 under the joint sponsorship of the Air Force Office of Scientific Research, the Army Research Office, and the Office of Naval Research. The Symposium was called primarily to determine the research required to achieve a major reduction in software costs because the software art was progressing slowly and high software costs and poor quality were having a serious effect on the DoD budget and operations.

The Symposium, well attended by 97 persons, was divided into five workshops, each of which was concerned with a specific aspect of software. The members of each workshop were assigred on a permanent basis (for the duration of the Symposium), and the workshops met simultaneously four times during the 3-day period. "The third workshop session was held in the form of five open houses, so that the developing ideas of each workshop could be exposed to outside comment." Three meetings of the symposium as a group were held during which subjects of interest to the entire body were presented. A keynote speech on software costs and statements of objectives by the workshop chairmen were the subjects of the first meeting. The second meeting was devoted to software technology transfer, while the third meeting included interim progress reports by workshop chairmen.

The five workshops were assigned the following themes:

Workshop 1 — Understanding the Software Problem

Workshop 2 — Semantics of Languages and Systems

Workshop 3 — Programming Methodology

Workshop 4 — Software-Related Advances in Computer Hardware

Workshop 5 — Problems of Large Systems

10.2 FINDINGS/CONCLUSIONS

"Over the last ten years there has been a radical shift in the balance of hardware and software costs. Because of technological advances, hardware costs have been reduced to the point where hardware designers are now seeking ways to help reduce software costs. The cost of computing is now clearly dominated by the cost of software."

Although the demands for software production are increasing in volume and complexity, the progress in software technology has been very slow. "Such demands have clearly outstripped the technology, with very costly results. Production of new software products suffers great overruns in cost and delivery time, and quality is often deficient in correctness, modifiability, and transferability. The maintenance costs for old software products may be an order of magnitude larger than production costs, due to poor original design and production."

Because software is often a critical component in large systems, delayed delivery times or poor quality can cause problems with related high costs that can far exceed the original costs, regardless of how high these might happen to be.

"There is, further, much waste in programming and computing, resulting from poor matching of software and hardware. Thus, incompatibility between computers results in costly reprogramming or an inability to take advantage of the reduced computing costs of new hardware. Also, poorly designed primitive functions in hardware require repeated costly and error-inducing programming of basic computational functions.

"The high direct and indirect costs of software set an effective practical limit to the complexity and scale of realizable systems. Therefore, a major reduction in software costs (including the costs resulting from bad quality) could have a great impact on the practical capability of logistic, avionic, tactical, communication, and other vital systems."

Certain symptoms or causes manifest themselves in problems related to software and the acquisition process. There is a definite interaction of technical and managerial aspects of software, and there also exists a certain air of uncertainty about software because of these kinds of situations, poor management practices in production control, acceptance of all levels of programmer talent, failure to utilize available production tools, and "failure to provide modern and adequate hardware resources, both for programming and for program execution."

"Rational, controllable software production practice requires more systematic methods and tools than we now have for specifying and measuring properties of programs with respect to all pertinent qualities, such as correctness, performance, and modifiability. In addition, we need better understanding of the programming process in its technical, psychological, and social aspects. A large fraction of so-called management problems and problems of inadequate tools are actually symptoms of the lack of fundamental understanding about the very complicated set of issues called software."

Strong trends are evident that will increase the need for software which will, in turn, probably further affect the already high costs. More users and more varied computing systems lend further support to the fact that software needs will increase.

"The symposium revealed a large body of ideas for scientific study and technological development that have clear potential for major impacts on software practice. The expected benefits of the various ideas varied in time frame.

"One set of ideas was aimed at understanding and improving current modes of software practice. These include applying and refining the best current methods in documentation, debugging, testing, and production control. It was felt that certain techniques within the research community could be transferred immediately, with promise of excellent results.

"A second set of ideas was aimed at developing new software methodologies and improved computer architectures for applications poorly served by present systems. The new methodologies include aids to the programmer for understanding complex problems, for designing systems, and for analyzing program and system behavior. There are many attractive approaches that require intensive development effort.

"A third set of ideas was aimed at making programming a more automatic process, both for expert and nonexpert programmers, and for improving computer system design. These ideas require long-range development, but some early work is needed to guide evolving practical techniques in programming and computer architecture.

"The chairmen's summaries and recommendations reveal a deep sense of urgency reflecting a widespread feeling of the workshop members. Together with this feeling there was a conviction that good ideas are available that can be expected to have strong impact on software practice if pursued energetically.

'While the ultimate success of the many particular ideas could not be predicted, there were ample cases of partial success to justify a high feeling of confidence among the attendees.

"In summary, present direct and indirect software costs constitute a serious limit to the capabilities that can be achieved in systems operated by the services. Future software demands are visible whose character and scale will greatly increase the services' software costs. A strong program to advance the software art is therefore urgently needed."

10.3 RECOMMENDATIONS OF THE MONTEREY SYMPOSIUM ON THE HIGH COST OF SOFTWARE

1. Research in computer systems should be strengthened and closely coupled to software research.

2. Because software research has tended to separate basic programming methods from application programming, it is becoming clear that knowledge from a particular application domain is needed in order to increase the power of programming aids for that domain.

3. Many powerful software techniques, now in laboratories, should be immediately transferred to actual users to enhance their productivity and knowledge.

4. There should be a strengthened technology base, available directly to the services.

5. Service-supported research should be coordinated with other DoD and civilian research and development.

6. The scale and quality of computer research should be increased to meet present and future demands.

7. An understanding of software costs should be developed.

8. The use of the best available programming aids, e.g., for program writing, documentation, debugging and management, should be integrated, applied, and evaluated.

9. The theory and practice of structure-oriented programming methodology should be developed.

10. New concepts for program testing and analysis should be developed.

11. Human factors in programming and computer utilization should be investigated.

12. New programming methods and improved computer architectures for major new application areas should be developed.

13. Concepts and techniques for realizing knowledge-based systems for important application domains should be developed.

14. Both theory and effective methods for formal verification and proof of program properties should be developed.

15. The semantic basis for constructing better programming languages and computer systems should be improved.

10.4 CORRELATION WITH APL RECOMMENDATIONS

The findings and recommendations from the Symposium on the High Cost of Software correlate most closely with the following four APL recommendations: SE1, IP1, IP2, and TT1.

1. System Engineering of Computer Systems SE1

   For systems involving several distinct functions, require that the system be divided into functional segments in accordance with the operational requirements. Require during the Program Validation Phase that tradeoff analyses be performed for hardware versus software (i.e., hardwired versus programmable functions) and for different computer system architectures.

2. Software Development Support Tools and Facilities IP1

   Ensure that the Full Scale Development program includes provision for adequate modern support tools and facilities, including such items as assemblers, compilers, editors, debugging aids, data base and library management systems, and associated operating systems. Require maximum use of existing proven tools and facilities. Provide that any of these that will be required by the Operational Support (Maintenance) Agent for system maintenance be delivered in transferable form and also be capable of application to future Weapon System programs.

3. Disciplined Programming IP2

   Require that the computer program development contractor apply a highly disciplined set of engineering practices to the detailed design and programming phases of development. This must involve a clear and disciplined set of standards covering program structure, size, control, interface, formal conventions on data base management, and the demonstration that the standards are enforced in practice.

4. Software Test Tools TT1

Support development of improved software test and validation tools to reduce the cost and time involved in software verification. These should include automated tools to identify and exercise all branches, to detect and isolate design faults, and to categorize error sources.

GOVERNMENT/INDUSTRY SOFTWARE SIZING AND COSTING WORKSHOP

11. GOVERNMENT/INDUSTRY SOFTWARE SIZING AND COSTING WORKSHOP

11.1 INTRODUCTION

The Government (or Electronics Systems Division, ESD)/Industry Software Sizing and Costing Workshop was held on 1-2 October 1974, at the Air Force Systems Command (AFSC) ESD, Hanscom Air Force Base. Massachusetts. The workshop was well attended with approximately 76 attendees of whom half were Government representatives and the other half were from industry. Twenty-one companies, one university, and nine USAF and Governmental units were represented.

The general purpose of the workshop was to seek a means of enhancing communications between the Government and industry on the problems of predicting software development costs. "More specifically, the workshop focussed attention on two key questions.

"What are the attributes of a good software requirements specification?

"What are the prime factors affecting/driving software costs?"

The ultimate objective was to enhance significantly the realism/credibility of future software costing and sizing estimates for electronics defense systems.

In order to have discussion groups of workable size, the workshop was divided into four splinter groups of approximately 20 people each. The small groups addressed the two questions stated above and developed answers that are summarized in the Draft Report dated 11 February 1975.

11.2 FINDINGS/CONCLUSIONS

The workshop participants arrived at several important conclusions. A sampling of these conclusions is as follows:

1. "The purpose of a specification is to communicate and record the requirements of a system/project throughout its life cycle. Typically, the system life cycle in terms of specs may look like the following:

Conceptual - - Required Operational Capability

Development - - Requirements Analysis and Validation
doc. Type A, System Level Spec.

Production/
Acquisition - - Type B, Development Specs.

Type C, Product Specs.

Operation/
Maintenance - - Previous doc. as updated plus others

The level of detail to which the foregoing list of representative specifications addresses software varies greatly as one progresses through the development cycle."

2. "The group noted a 'giant void' in that a thorough requirements analysis and validation is not presently performed in many cases prior to writing the system spec."

3. "Often critical performancc goals or permissible trade-offs affecting the software design are not revealed in the RFP specifications. This leads to widely varying bid estimates and in general makes it less likely that each responder will produce the best proposal and design of which he is capable. In response to the same RFP it is common to have a five to one ratio in bids for software efforts."

4. "Another example of factors affecting a software design that may be omitted from an RFP is the failure to specify the maximal, minimal, and nominal expected operating conditions and the performance required under such conditions."

5. "A significant problem cited was that RFPs sometimes specify a design instead of performance requirements to be met by a design."

6. "Separating design ideas from performance requirements is in the government's best interest; it makes it more likely that the benefits of improved design ideas can be obtained in procured systems."

7. "One point which found very little argument; to derive a good software cost estimate is very expensive. It was generally agreed that in order to accurately predict software costs for a project, one must do a considerable amount of design work in addition to project planning."

8. "One concept which the independently guided panels or splinter groups seemed to view with favor was the possibility of a phased contract approach or even a separate contract altogether for software development effort."

9. "The number of delivered executable source instructions is currently the most widely-used factor for cost estimation."

10. "In general, participants' experience indicated that the cost per source instruction in assembly language or machine-oriented language (MOL) was about twice the cost per source instruction in a higher-order language (HOL) such as COBOL or FORTRAN. The dollar figures were derived from an estimate of 15-30 HOL source instructions/man-day and the typical figure of $35,000 per burdened man-year for software manpower."

11. "Some attempts have been made to correlate costs with such factors as number of interfaces, percentage of branch statements, number of paths through a program, and Halstead length, but so far without any highly reliable correlations."

11.3 RECOMMENDATIONS OF THE GOVERNMENT/INDUSTRY SOFTWARE SIZING AND COSTING WORKSHOP

1. A possible multiphased (definition, production) and/or separate contract approach to software acquisition should be considered.

2. Software specification standards and practices need much improvement to ensure consistency, proper level of detail, and clear conveyance of minimum requirements.

3. There is a need to initiate standard terminology, improved work breakdown structure, and collection of good historical cost data.

4. The fact should be emphasized that there are almost no shortcuts to deriving a good software estimate. The process itself will continue to be very costly since significant software design effort must be expended.

11.4 CORRELATION WITH APL RECOMMENDATIONS

The findings and recommendations of this Government/Industry Software Sizing and Costing Workshop correlate most closely with the following APL recommendation: AM1.

1. Standard Criteria for Weapon System Computer Resources Acquisition Management AM1

   Establish a common set of requirements and criteria to be applied in the acquisition and support of Weapon System computer resources by all services.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

The work reported in SR 75-3A was done under Navy Contract N00017-C-72-4401. This work is related to Task ZC-6, which is supported by NAVSEASYSCOM.

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF DEFENSE			
ARPA	Washington, DC		1
DoD Software Management Steering Committee	OSD (I&L), Pentagon Room 2A318		25
Defense Systems Management School	Washington, DC		1
DDC	Alexandria, VA		12
Department of the Navy			
NAVSEASYSCOM	Washington, DC	6541	1
		6542	1
		PMS-378	1
		PMS-393	1
		PMS-403	1
NAVAIRSYSCOM	Washington, DC	533	2
		5331	1
		PMA-231A	1
		PMA-2'1VM	1
NAVELEXSYSCOM	Washington, DC		2
		PME-108	1
NAVMATSYSCOM	Washington, DC	PM-2	1
NAVSEC	Washington, DC	6177	1
FCDSSA(DN)	Dam Neck, VA		2
FCDSSA(SD)	San Diego, CA		2
NADC (Naval Air Development Center)	Washington, DC		1
NMC	Pt. Mugu, CA		1
NSWC (Naval Surface Weapons Center)	Washington, DC		1
NUC (Naval Undersea Center)	Washington, DC		1
NUSC (Naval Underwater Systems Center)	Washington, DC		1
NSWSES	Port Hueneme, CA 93043	Code 3400	1
		Code 1223	[illegible]
Department of the Army			
MICOM (Army Missile Command)	Redstone Arsenal Huntsville, AL 35809	SAM-D	2
MICOM	Redstone Arsenal Huntsville, AL 35809	Pershing	2
BMDSC (Ballistic Missile Defense Systems Command)	Washington, DC		1
CENTACS, Army Material Command	Fort Monmouth, NJ		1
Department of the Air Force			
AFSC/XRF (Air Force Systems Command)	Headquarters, Andrews AFB Washington, DC 20334	SRWG	1
RADC (Rome Air Development Center)	Washington, DC		1
CONTRACTORS			
Boeing	Mail Station 8782 P. O. Box 3999 Seattle, WA 98124	Lee Campbell Software and Computer Sys. Eng. Dept.	1
Electric Boat, General Dynamics Corp.	Groton, CT 06240	Supervisor of Ship Building	1
General Research Corp.	6300 Hollister Ave. P. O. Box 3587 Santa Barbara, CA 93105		1
GRD, Inc.	Warminster, PA		1
Grumman Aero Space Corp.	Bethpage, NY 11714		1
Hughes Aircraft Company	Fullerton, CA 92634		1
IBM	18100 Frederick Pike Gaithersburg, MD 20760	Trident	1

Requests for copies of this report from DoD activities and contractors should be directed to DDC, Cameron Station, Alexandria, Virginia 22314 using DDC Form 1 and, if necessary, DDC Form 55.

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group

!NITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

SR 75-3A

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
CONTRACTORS (cont'd)			
ITT	7821 Orion Ave. P. O. Box 7713 Van Nuys, CA 91409		1
Lockheed Aircraft Corp.	Burbank, CA 91503		1
Martin Marietta Aerospace	P. O. Box 5837 Orlando, FL 32805	Pershing	1
Planning Research Corp.	McLean, VA		1
RCA	Morristown, NY 08057	Government and Commercial Systems	1
Raytheon Company	Boston Post Rd. Wayland, MA 01778		1
Systems Development Corp.	Santa Monica, CA		1
TRW	One Space Park Redondo Beach, CA 90278	Systems Engineering and Integration Division	1
Univac	P. O. Bcx 3525 St. Paul, MN		1
Vitro Laboratories (Automation Labs)	14000 Georgia Ave. Silver Spring, MD		1
UNIVERSITIES			
University of Maryland	College Park, MD		1
University of Sourthern California	Marina Del Rey, CA	USC-ISI	1

*Initial distributio of this document within the Applied Physics Laboratory has been made in accordance with a list on file n the APL Technical Publications Group.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

PLEASE FOLD BACK IF NOT NEEDED
FOR BIBLIOGRAPHIC PURPOSES

14

REPORT DOCUMENTATION PAGE

1 REPORT NUMBER
APL/JHU-SR-75-3A

2 GOVT ACCESSION NO

3 RECIPIENT'S CATALOG NUMBER

6

4 TITLE (and Subtitle)
DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY. APPENDIX A. FINDINGS AND RECOMMENDATIONS OF PREVIOUS STUDIES.

9

5 TYPE OF REPORT & PERIOD COVERED
Special Report.

6. PERFORMING ORG. REPORT NUMBER

10

7 AUTHOR(s)
J. M./Park

15

8 CONTRACT OR GRANT NUMBER(s)
N00017-C-72-4401 ✓

9 PERFORMING ORGANIZATION NAME & ADDRESS
The Johns Hopkins University Applied Physics Laboratory ✓
Johns Hopkins Road
Laurel, MD 20810

10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Task ZC-6

11 CONTROLLING OFFICE NAME & ADDRESS
Office of the Assistant Secretary of Defense
(Installations and Logistics), B. C. DeRoze
Pentagon, Rm 2A318, Washington, DC 20301

11

12 REPORT DATE
June 1975 ✓

13 NUMBER OF PAGES
92

14 MONITORING AGENCY NAME & ADDRESS
Naval Plant Representative Office
Johns Hopkins Road
Laurel, MD 20810

12 109p.

15 SECURITY CLASS (of this report)
Unclassified

15a DECLASSIFICATION/DOWNGRADING SCHEDULE
NA

16 DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18 SUPPLEMENTARY NOTES

19 KEY WORDS (Continue on reverse side if necessary and identify by block number)
Computer Software
Disciplined Programming
Software Acquisition Guides
Software as Contract Deliverable
Software Development Support Tools and Facilities
Software Operational Support Agent
Software Test Tools

20 ABSTRACT (Continue on reverse side if necessary and identify by block number)

This appendix to the DoD Weapon Systems Software Management Study conducted by APL contains background material extracted and/or summarized from 10 previous DoD-sponsored studies. The studies were designated Baseline Documents by the Department of Defense Software Management Steering Committee and are particularly relevant to the subject of Weapon Systems software. A brief introduction specifying the purpose of each study and a summary of its findings and/or conclusions are included. Recommendations are summarized for each study that provided them. Whenever such study recommendations are available, abbreviated versions of the APL recommendations (from the main report) that correlate most closely are included for reference.

Unclassified 031650

SECURITY CLASSIFICATION OF THIS PAGE